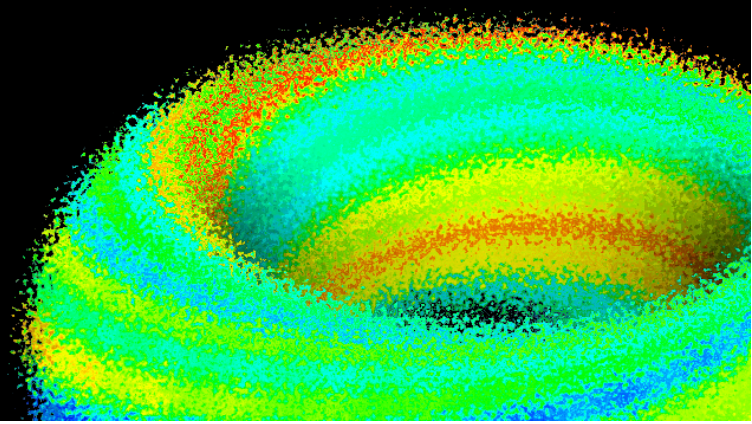
Supercomputing for Fusion Energy Applications

Fusion Group



Barcelona Supercomputing Center
Centro Nacional de Supercomputación

Xavier Sáez Pous

February 18, 2016

Outline

Introduction to Plasma Physics

Supercomputing for Fusion Energy Applications

Particle-in-Cell Codes

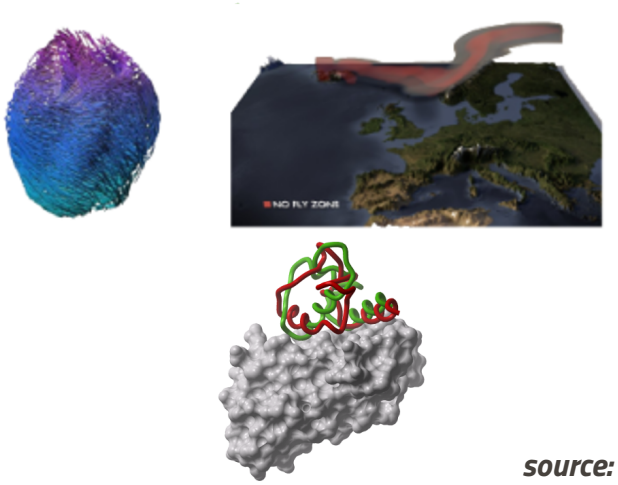
Fusion Group

Outline

- Introduction to Fusion Energy
- Supercomputing for Fusion Energy Applications
- Particle-in-Cell Codes
- Fusion Group

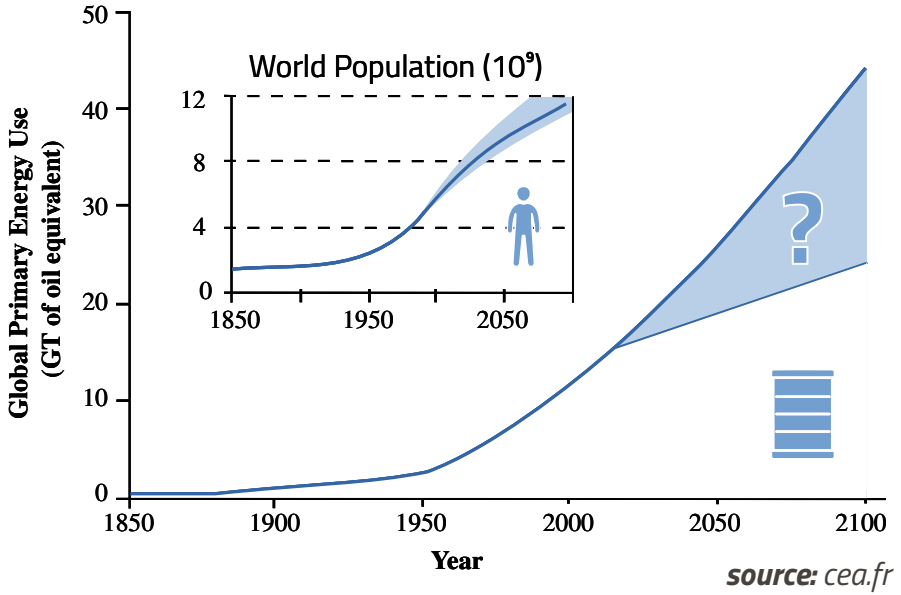
Motivation

- Grand scientific challenges are the driving force for High Performance Computing (HPC) evolution

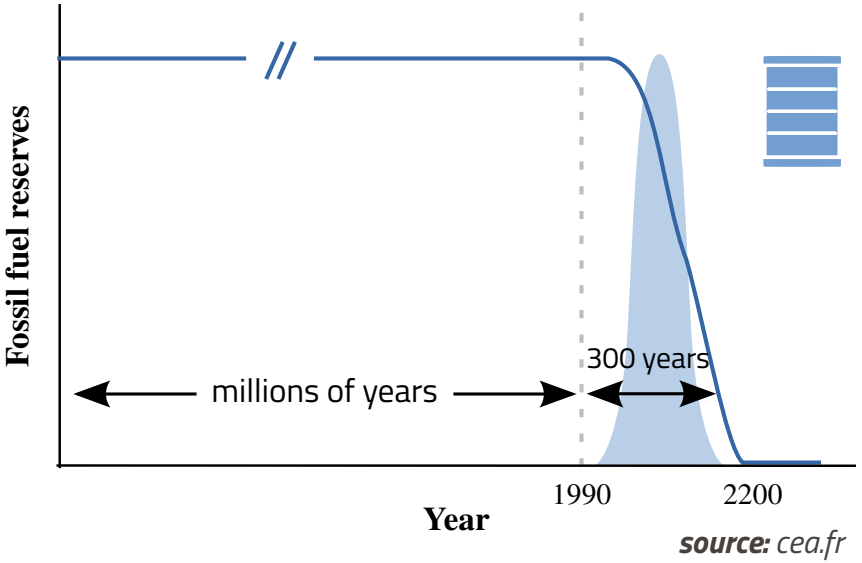


source: bsc.es

- Future energy needs



source: cea.fr



source: cea.fr

Objective

It is considered a key sector, e.g.

EUROfusion, EUFORIA...

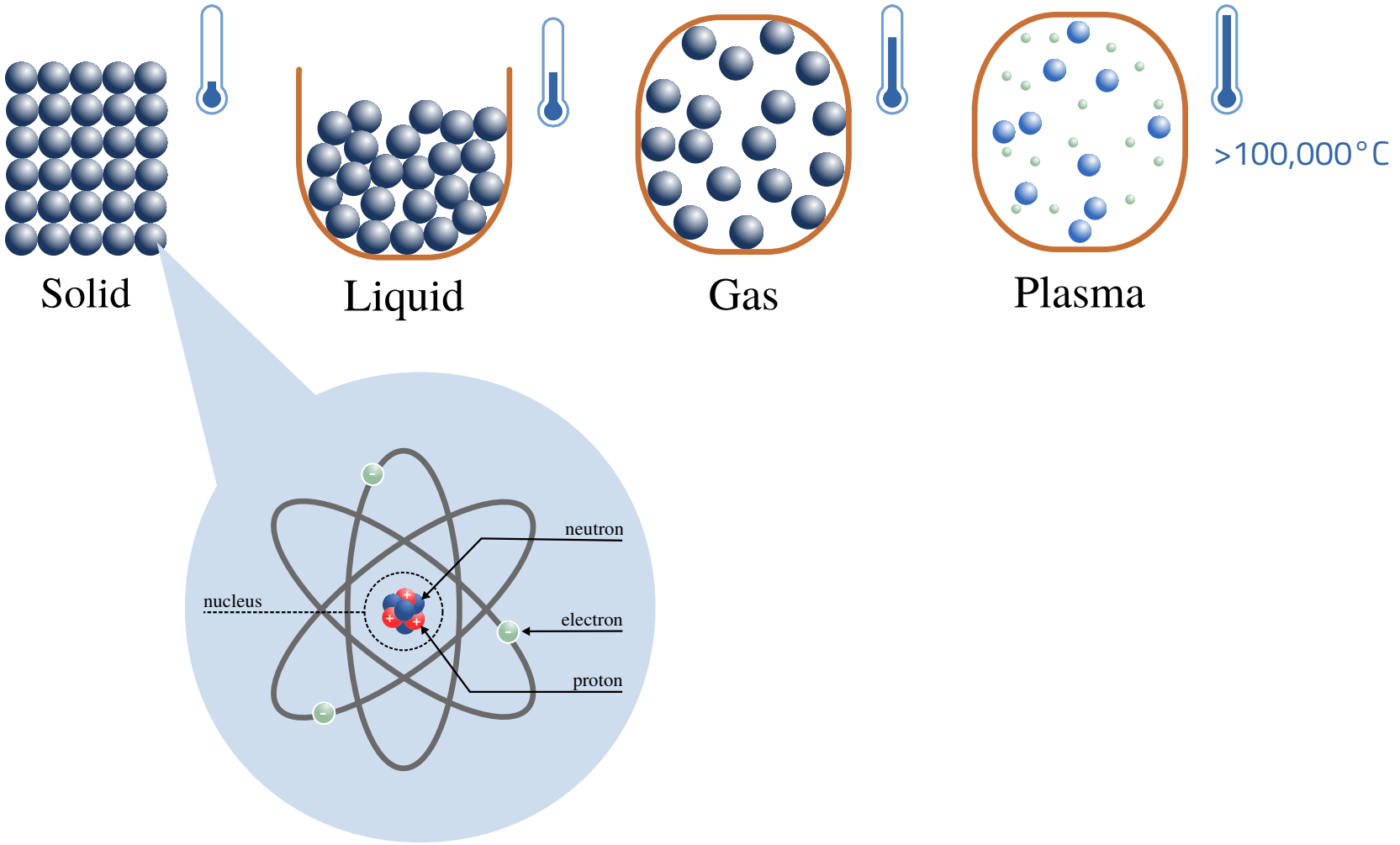
Context

- Plasma physics (from 1950's)
  - experiments are difficult and expensive (e.g. fusion reactors) → computational simulations
  - $\geq 10^{20}$ particles / m^3 (tokamak) → require intensive computation
- Plasma codes
  - codes suffer from the defects of dealing with idealized configurations due to lack of computing power → process level parallelism
  - for new fusion reactors (ITER) → more physics and detail → need to adapt to new platforms

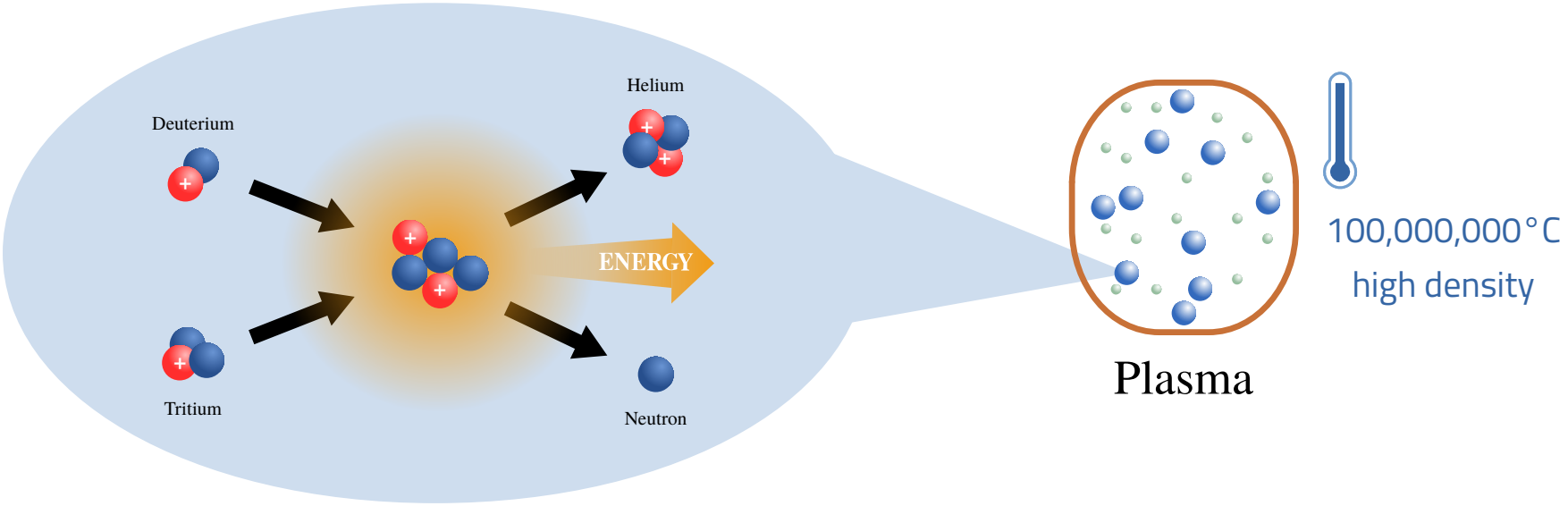
Context

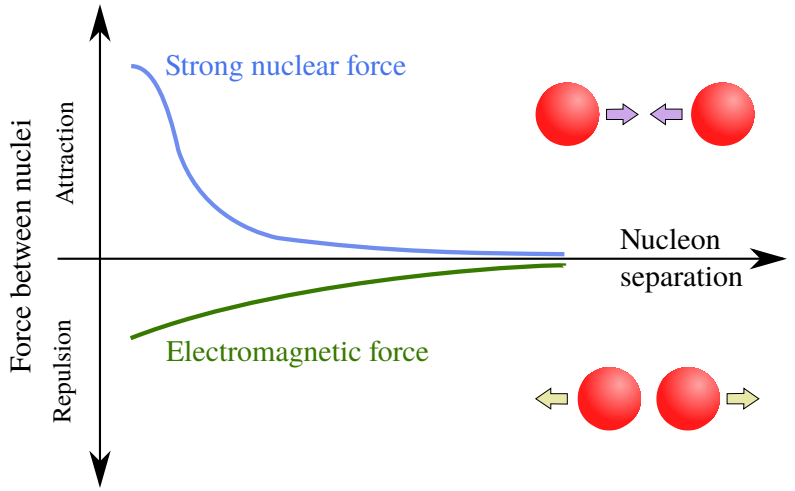
- High-Performance Computing (HPC) has evolved from single-core architectures
  - power efficiency has become a primary concern
    - multi-core architectures
    - Mont-blanc project → mobile processors
- If we want to add more physics and detail → we need to adapt plasma codes to new platforms

- What is Plasma?



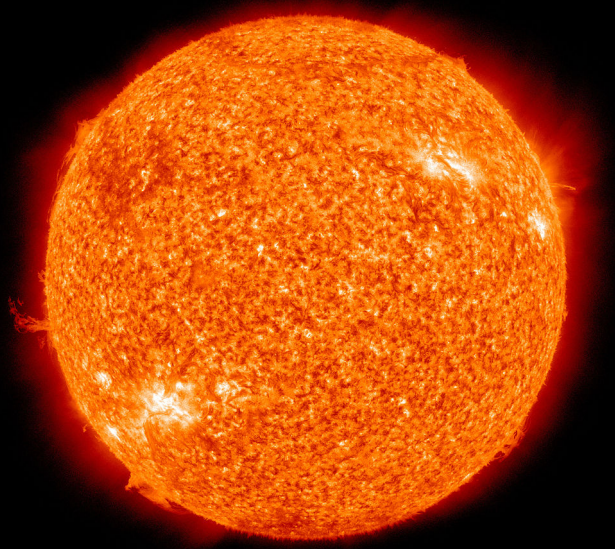
- What is Nuclear Fusion ?





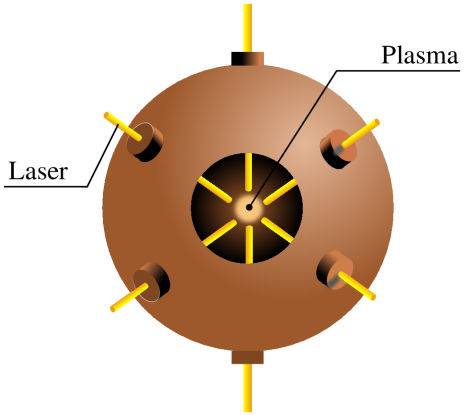
- Fusion Reactors (Space)
  - Gravitational confinement

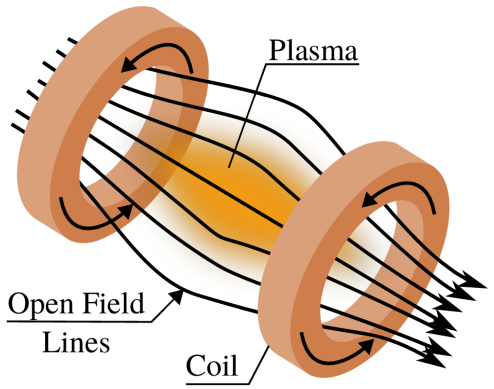
To confine the plasma to create the conditions necessary for fusion reactions to occur



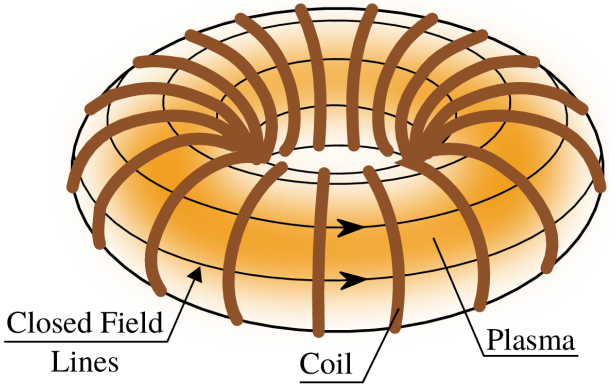
source: *nasa.gov*

- Fusion Reactors (Earth)
  - Inertial confinement
  - Magnetic confinement

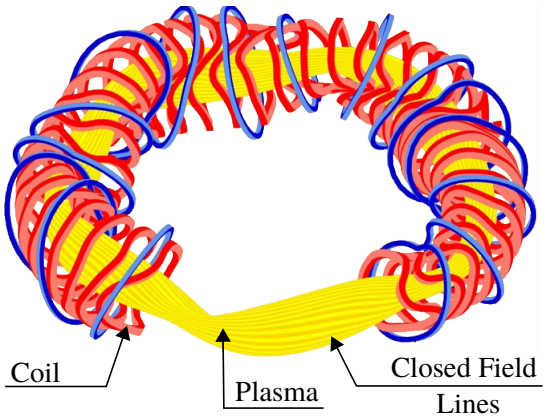




Magnetic Mirrors

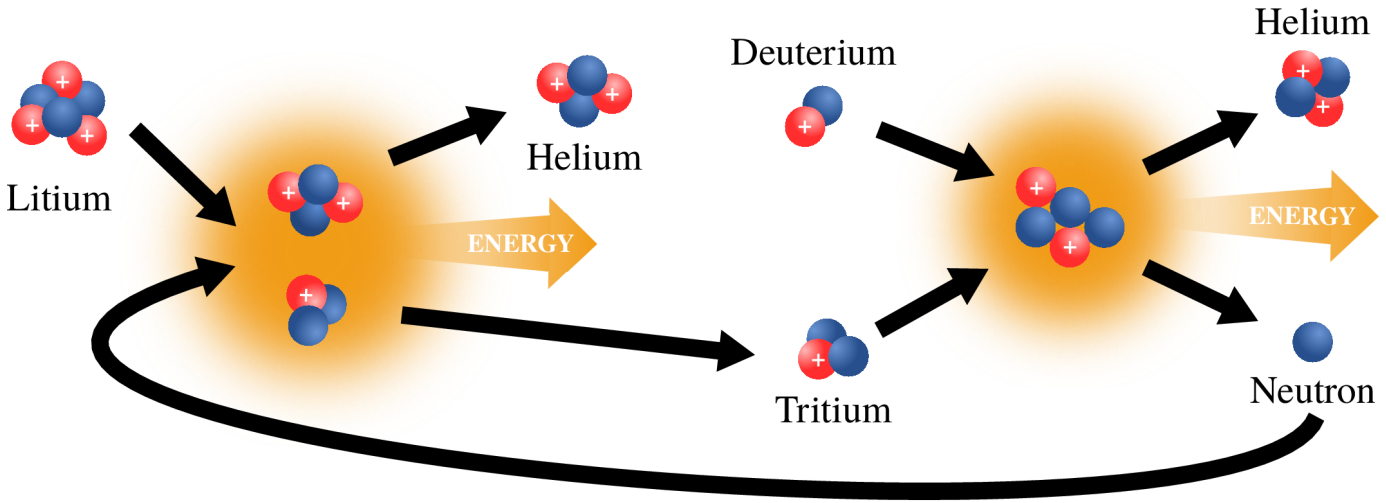


Tokamak



Stellarator

- Fusion Reaction
  - Deuterium-Tritium Fusion

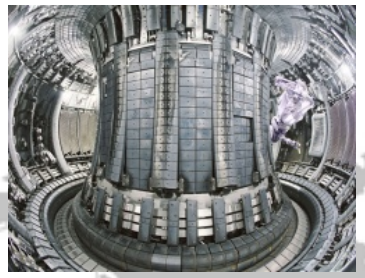


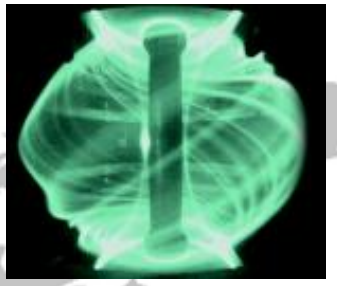
$$\mathrm{Li}^6 + n \rightarrow T + He^4 + 4.8MeV$$

$$\mathrm{D{+}T} \rightarrow He^4(3.5MeV) + n(14.1MeV)$$

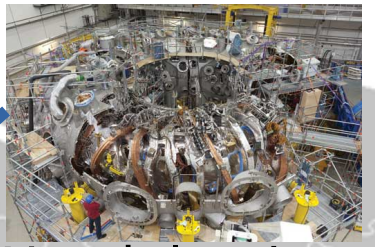
Fusion Reactors

- Fusion reactors in the world



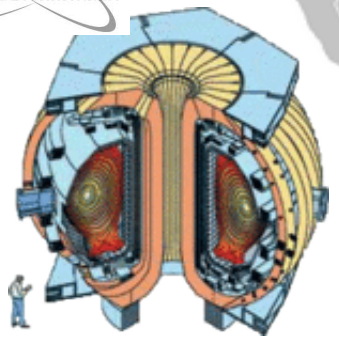


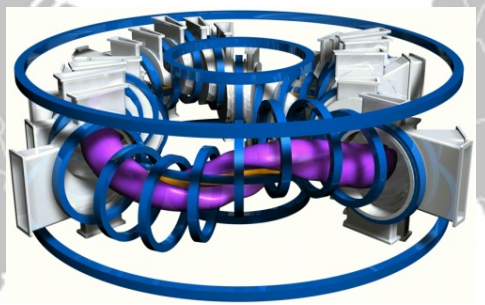
JET (EU) and MAST (UK)



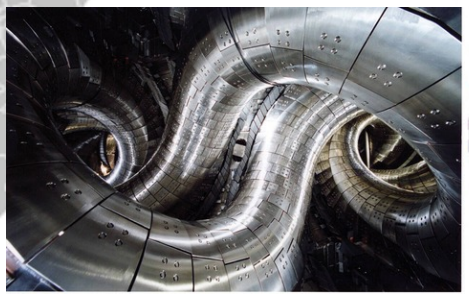
Wendelstein 7-X (Germany)

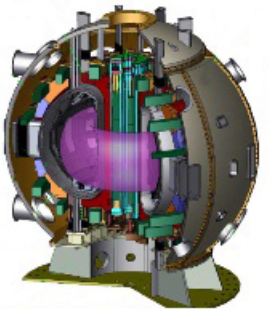
DIII-D NATIONAL FUSION FACILITY





TJ-II (Spain)

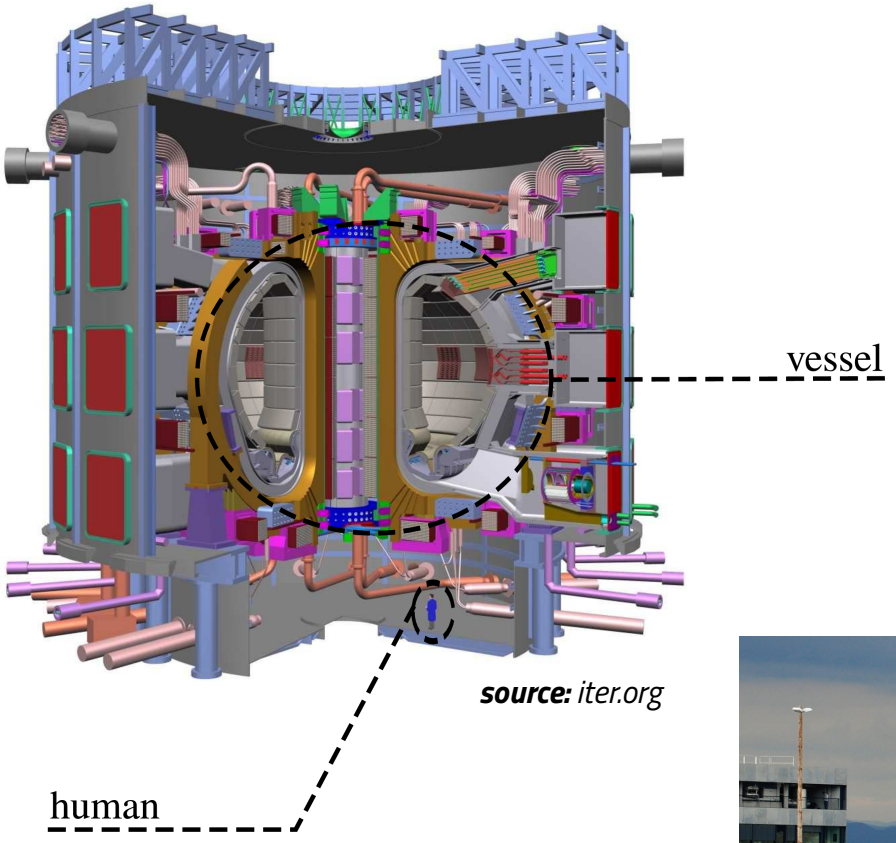




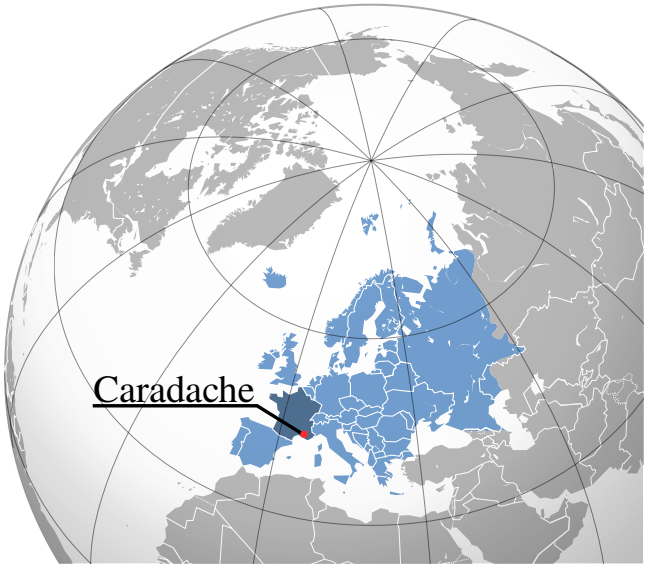
LHD and JT60U (Japan)

Fusion Reactors

- ITER *"The way"*



source: *iter.org*



source: *iter.org*

Fusion Reactors

- ITER *"The way"*

source: *The Daily Conversation*

- Maxwell's equations

Gauss's law
$$\nabla \cdot \vec{E} = \frac{1}{\varepsilon_0} \sum_{\alpha} q_\alpha \int f_\alpha d^3 v$$

Gauss's law for magnetism
$$\nabla \cdot \vec{B} = 0$$

Faraday's law
$$\nabla \times \vec{E} = -\frac{\partial \vec{B}}{\partial t}$$

Ampère's law
$$\nabla \times \vec{B} = \mu_0 \sum_{\alpha} q_\alpha \int \vec{v} f_\alpha d^3 v + \frac{1}{c^2} \frac{\partial \vec{E}}{\partial t}$$

- Poisson's equation

$$\nabla^2 \phi = -\frac{q}{\varepsilon_0} \int f\left(\vec{x}, \vec{v}, t\right) d\vec{v}$$

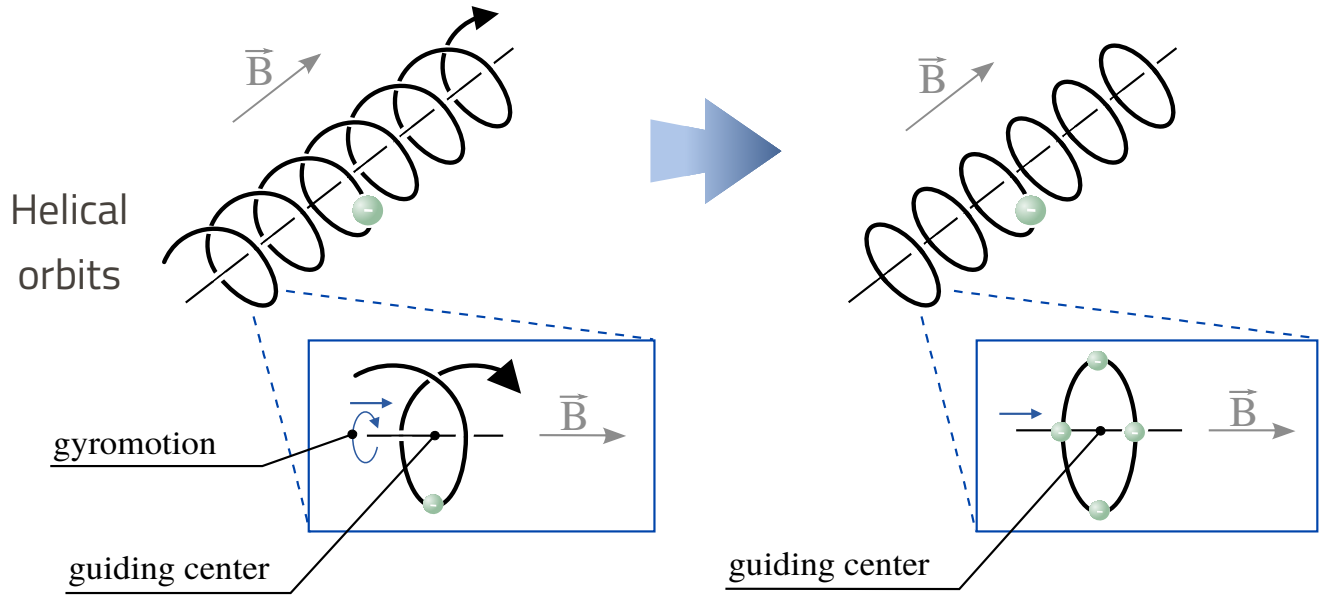
Plasma Physics

- Transport equation (6d+t)

$$\frac{\partial f_\alpha}{\partial t} + \vec{v} \cdot \frac{\partial f_\alpha}{\partial \vec{x}} + \frac{\vec{F}^{tot}}{m_\alpha} \cdot \frac{\partial f_\alpha}{\partial \vec{v}} = C_\alpha$$

collision-less
$C_\alpha \approx 0$

- Gyrokinetic approximation

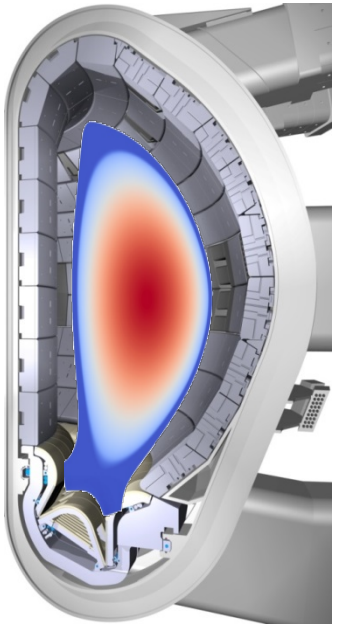


(5d+t)

Outline

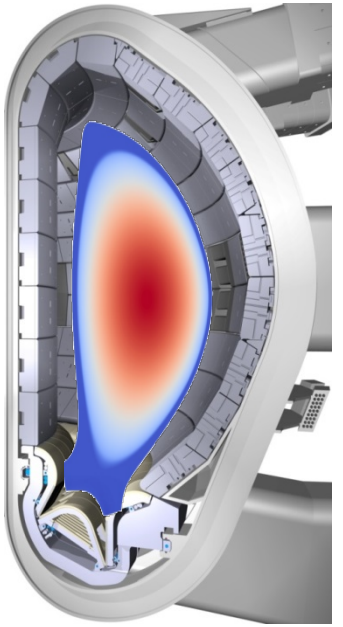
Big Questions

- How do we produce stable plasmas?
  - Can we feedback control it?
  - How do we sustain a fusion plasma in close proximity to a vacuum vessel?
- How do the particles and heat leak out?
  - How do we minimize the losses?
- How can turbulence finish confinement?
  - Could we trigger turbulence reduction externally?



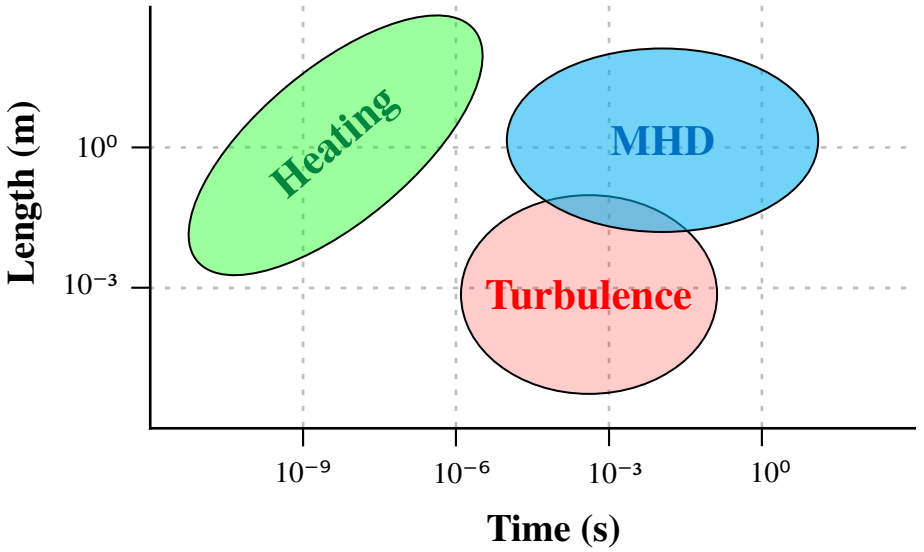
Big Questions

- We need fusion modelling to understand and control the plasma dynamics in tokamak and the implications for ITER:
  - Which materials do we use in fusion reactor?
  - How could we handle the intense flux of power, neutrons and charged particles on the wall?



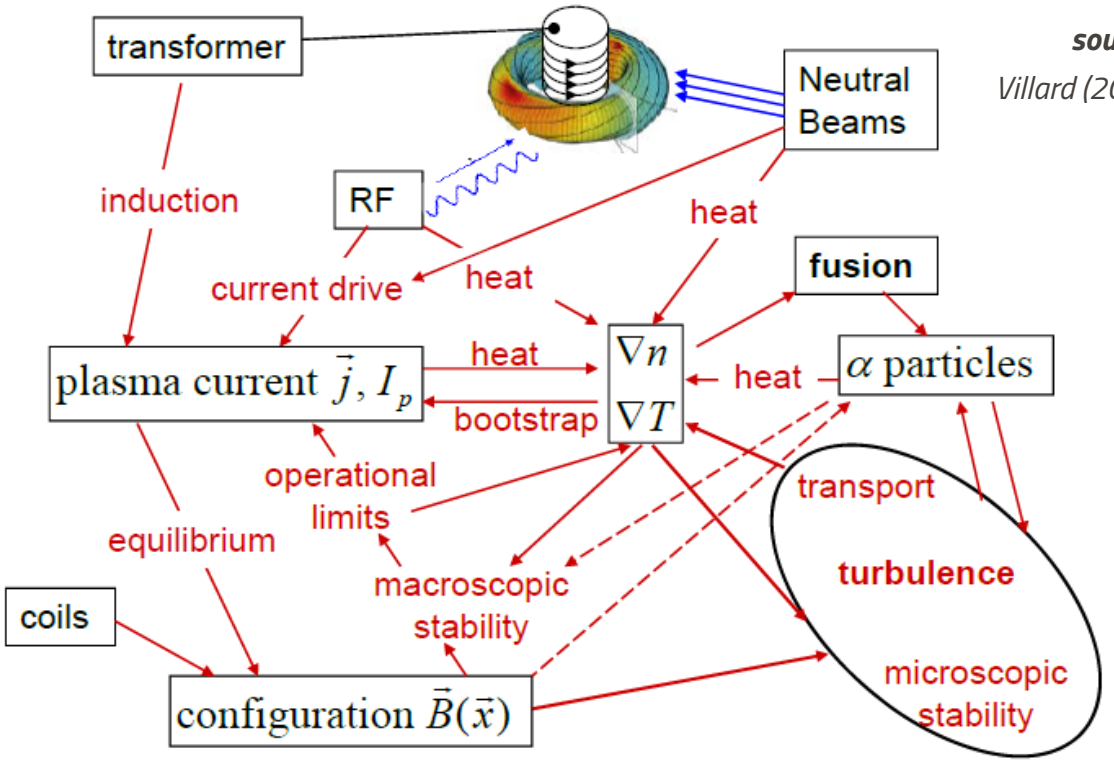
Fusion Modelling

- There are many non-linear phenomena in fusion plasmas
  - Energetic ions
  - Core transport
  - Transport Barrier
  - Non-inductive CD
  - RF Heating
  - NBI Heating
  - Neutral Transport
  - MHD stability
  - Impurity Transport
  - Turbulent Transport
  - Plasma Wall Int.
  - Divertor Plasma
  - SOL Transport
  - and more …
- Wide ranges of time scale and spatial scale interact each other.
  - Time scale: 10^{-12} s ~ 1000s
  - Spatial scale: 10um ~ 10m



Fusion Modelling

- Unfortunately, these phenomena can not be modelled in a single simulation in a reasonable time.



source: *Villard (2010)*

- Therefore, supercomputing is needed to simulate
  - Integrating relevant phenomena combining modelling codes in a consistent manner
  - Dealing with various levels of physics model

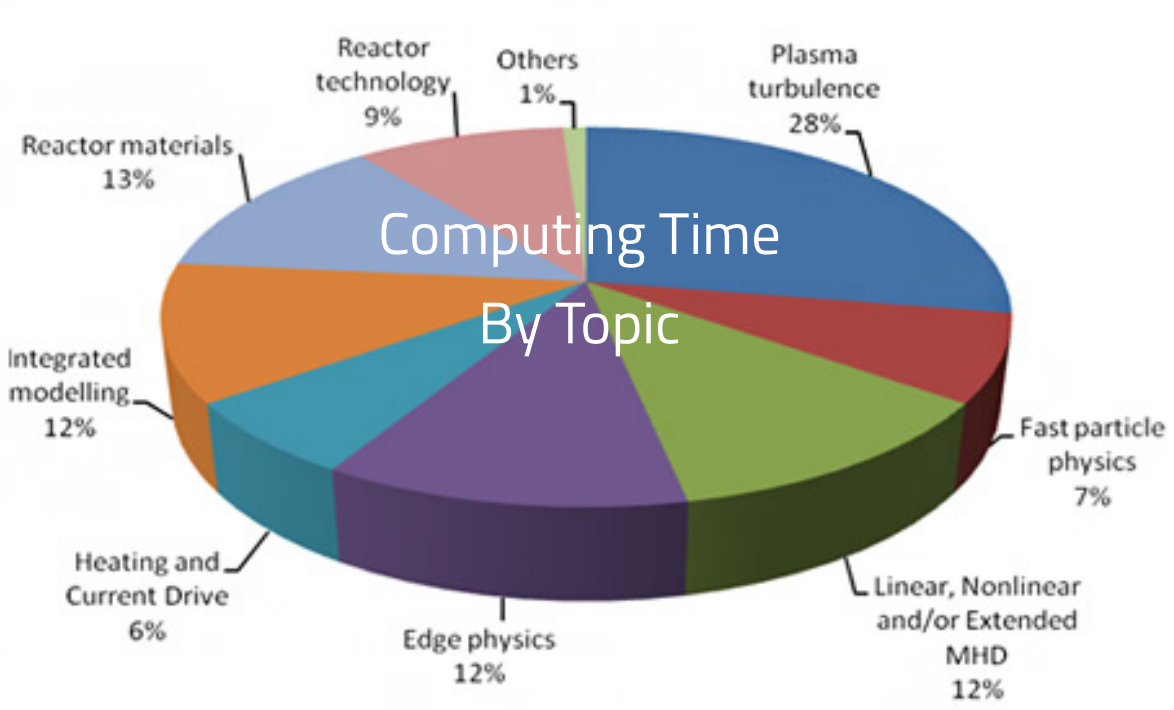
Supercomputing

- To sum up, it allow us to face these problems:
  - to analyze experimental data on fusion plasmas
  - to improve understanding of the physics involved and to validate the theoretical models and the modelling codes
  - to prepare scenarios for ITER operation
  - to predict the performance of the ITER facilities
  - ...

Supercomputing

- Example: **HELIOS** supercomputer
  - Located in Rokkasho (Japan)
  - Only for fusion simulation projects
  - 4410 bullx B510 compute nodes
  - Each node contains 2 Intel 8-core Sandy Bridge EP processors with 58 GB of RAM
  - 1.52 PFLOPS

The HELIOS Supercomputer system at IFERC-CSC.

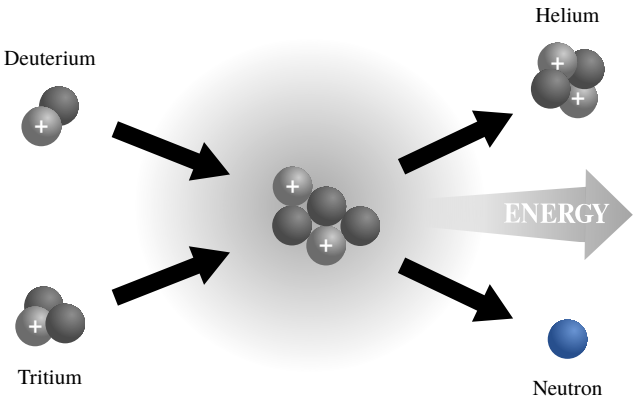


Modelling of Fusion Plasmas

- Fusion neutronics
- Fast particle physics
- MHD in fusion plasma
- Plasma turbulence

Modelling of Fusion Plasmas

- Fusion neutronics
  - Fusion plasma is a source of neutrons which impinge on all material surrounding the plasma.
  - Through interaction with neutrons, materials become activated and subsequently, emit decay photons.



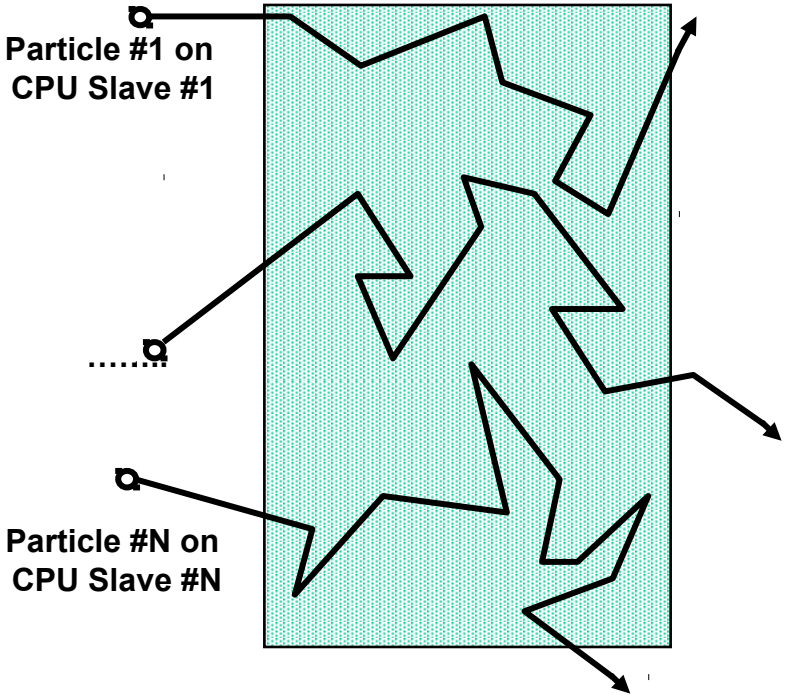
  - The resulting shut-down dose rate is an occupational health issue.

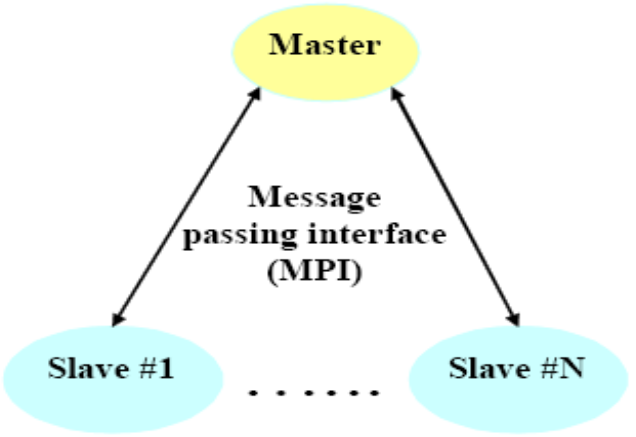
Two families of neutronics codes:

- **Deterministic**: PARTISN, DOORS, DEVONO, ATTILA,..
- **Monte Carlo**: MCNP, TRIPOLI,...

Modelling of Fusion Plasmas

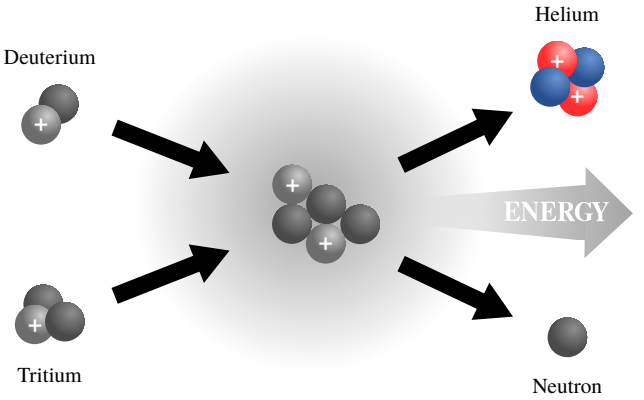
- Fusion neutronics
  - Monte Carlo codes
  - Tracking of individual particle pathways on microscopic level in complex 3D geometries of fusion devices





Productive parallel run on
slave processors

- Fast particle Physics
  - Different types of 'fast' particles (non-thermal) exist in a fusion reactor:
    - Helium-4 ions, neutral beam injected ions, particles heated by RF waves, ...



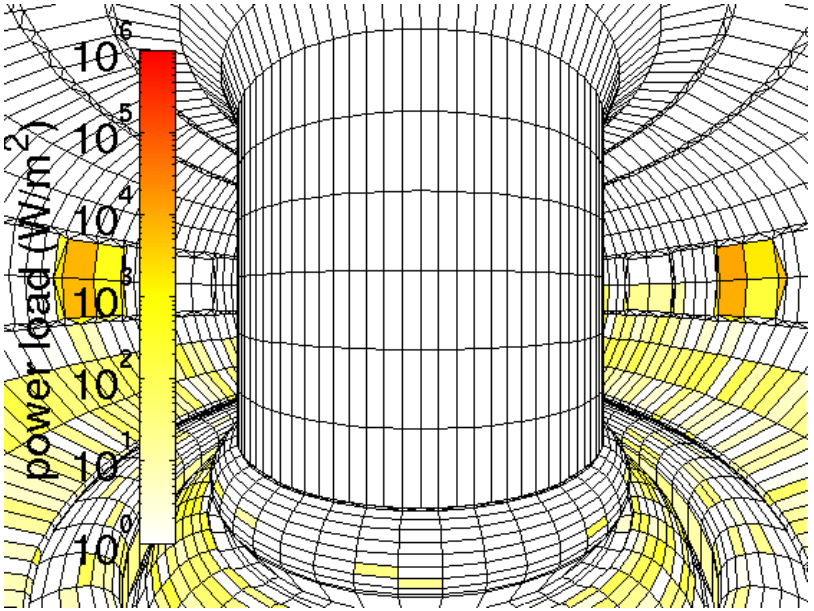
  - These particles will collide with other plasma particles. It can drive instabilities and possibly losses of fast particles can damage the surrounding materials and need to be kept at a tolerable level.

There are many codes:

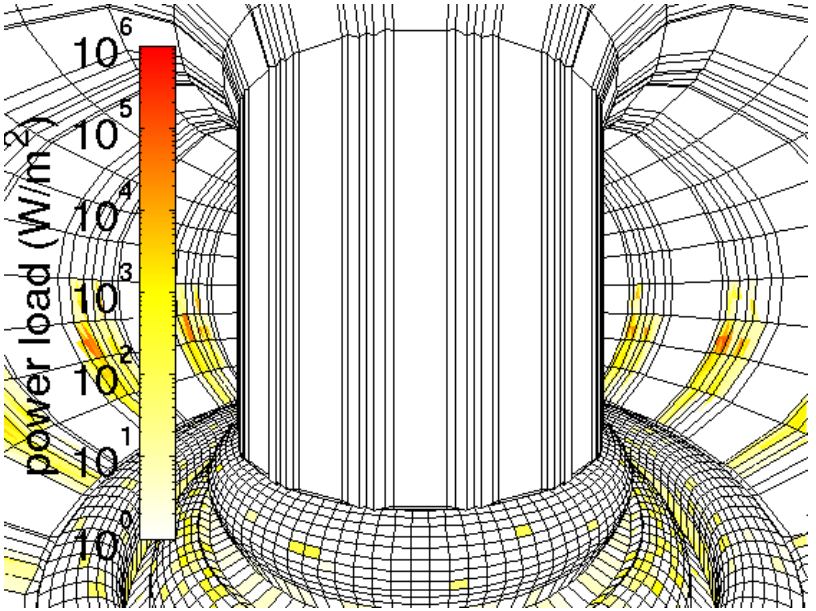
- OFMC-3D (Japan);
- SPIRAL (USA);
- DRIFT (Russia);
- ASCOT (Finland)...

Modelling of Fusion Plasmas

- Fast particle Physics
  - ASCOT: Racetrack for tokamak particles



Original wall w/ 2 limiters



Present wall design w/ poloidally extended 'continuous' limiters

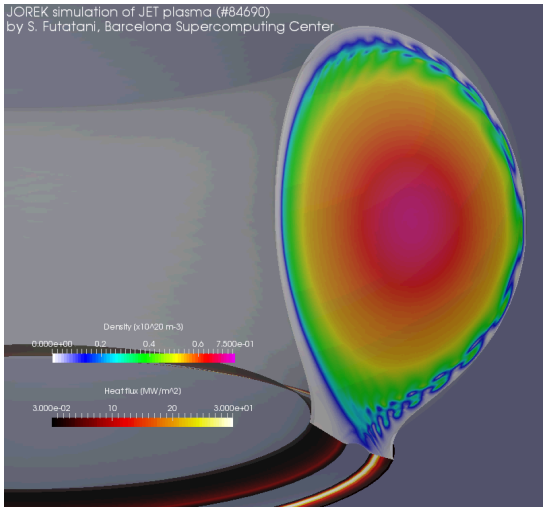
source: *Kurki-Suonio et al.*

Modelling of Fusion Plasmas

- Edge Localized Modes (ELMs)
  - A disruptive instability occurring in the edge region of a tokamak plasma
  - These instabilities can damage wall components, particularly divertor plates, due to their extremely high energy transfer rate
  - Safe ELMs for divertor:

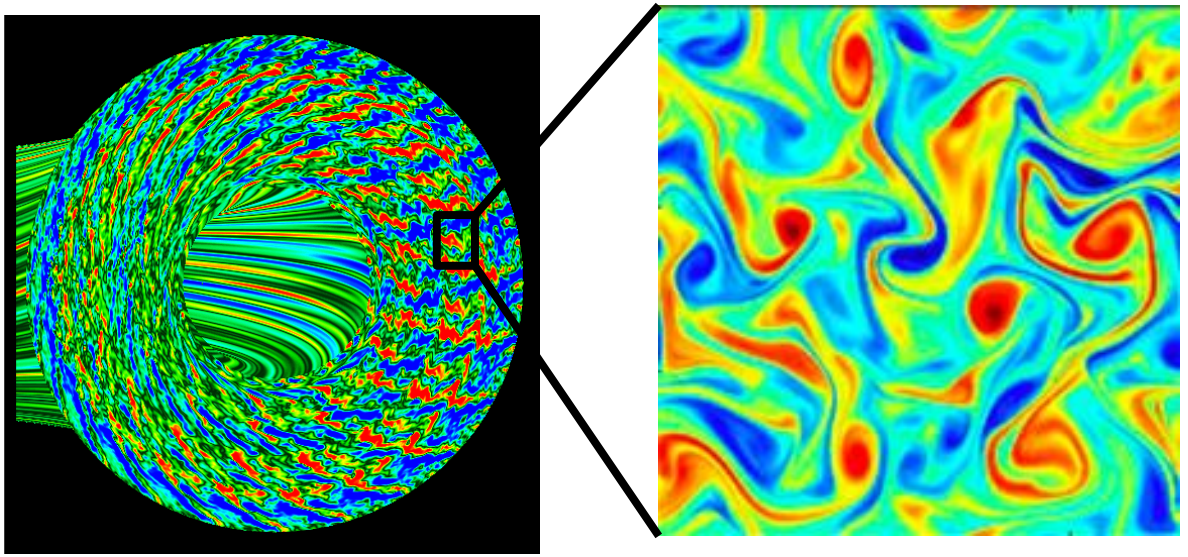
    $W_{ELM} < 1MJ$
  - Predictions for ITER:

    $W_{ELM} \sim 20MJ$



Modelling of Fusion Plasmas

- Plasma turbulence
  - Turbulence governs fusion plasma performance
  - Limits the maximum reachable density and temperature
  - Degrade the confinement



[Numerical Tokamak, Dorland et al.]

[Futatani et al.]

Outline

Introduction to Plasma Physics

Supercomputing for Fusion Energy Applications

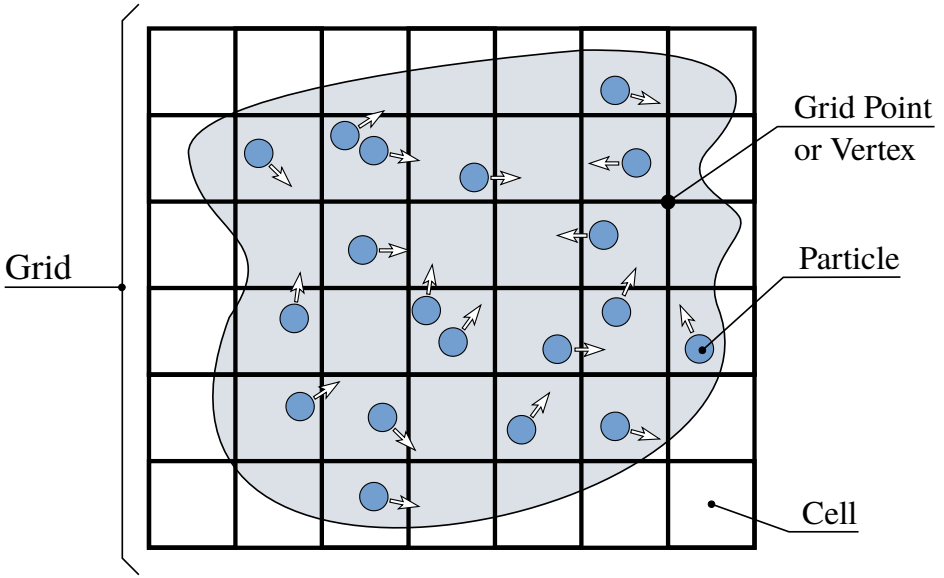
- Particle-in-Cell Codes

Fusion Group

Particle in Cell

- Particle-in-Cell (PIC) is a method to model physical systems whose behaviour varies on a large range of spatial scales.

- Two problems (trade-off) :
  - Macroscopic level, the dynamics is described by a continuum model
  - Microscopic level is modeled by a collection of discrete particles



Particle in Cell

- Algorithm

Initialize *particle* data

Initialize *grid* (field) data

while $t < t_{max}$ **do**

Compute particle contributions to the *grid*

Calculate the fields

Update forces on *particle* positions

Move *particles* to new positions

if display **then**

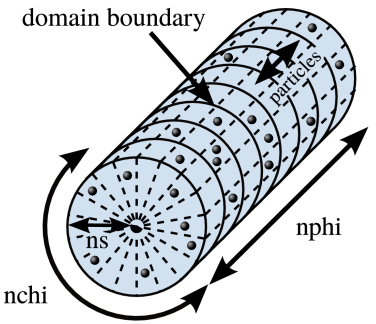
Print *particles* and *grid* data

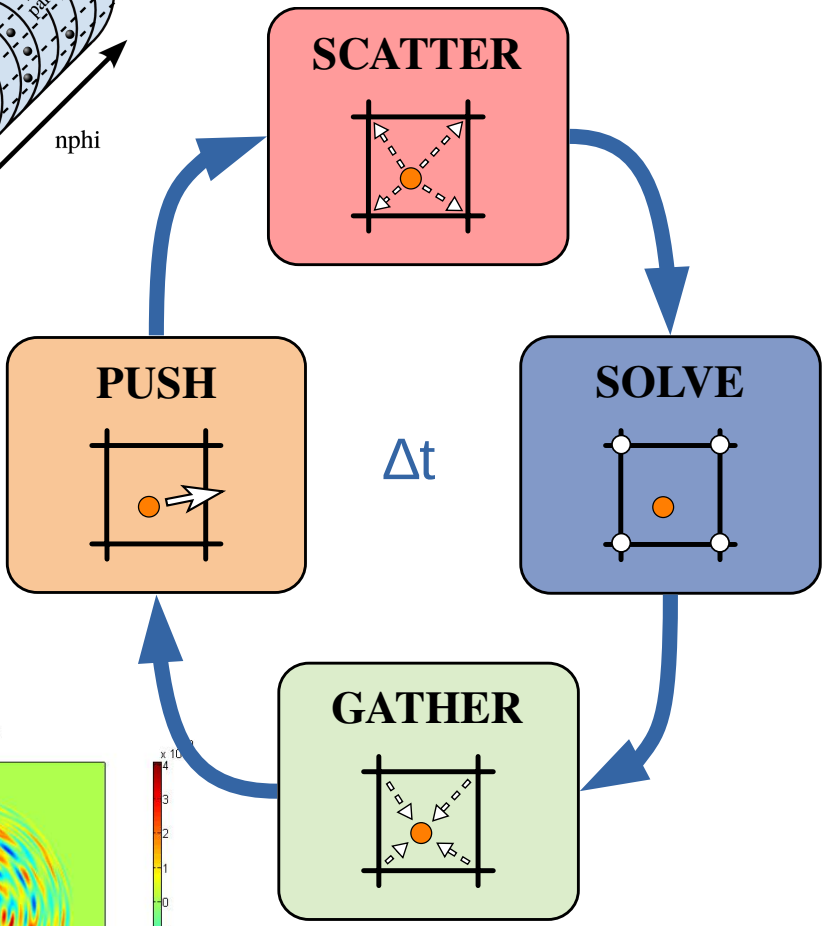
Print statistics

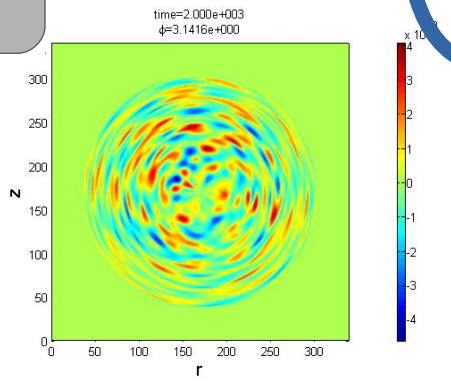
end if

$t \leftarrow t + t_{step}$

end while

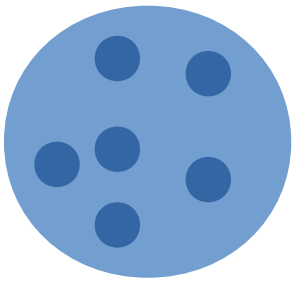


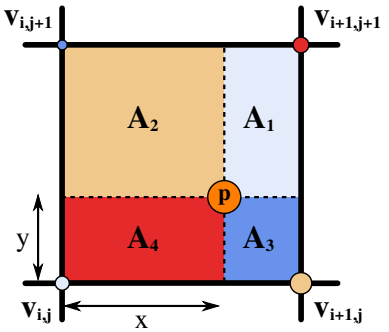


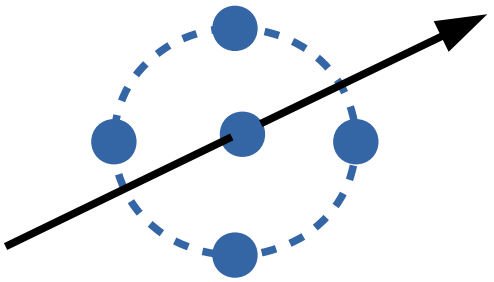


Particle in Cell

- PIC codes are very demanding in computational power
- Simplifications are introduced
  - Markers = *cloud of particles*
  - Linear interpolations (scatter and gather phases)
  - Collision-less
  - Gyrokinetic approximation: (6d+t) → (5d+t)



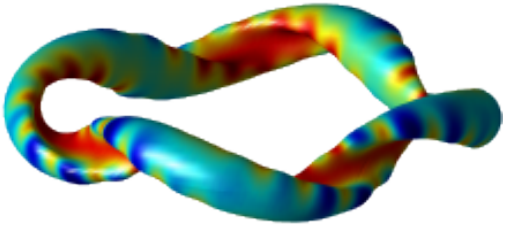




Particle in Cell

- Examples of Codes:
  - GENE
  - ELMFIRE
  - BIT1
  - EUTERPE
  - ...

- Gyrokinetic PIC code for the simulation of plasma microturbulence in 3D geometries (stellarators, tokamaks, etc)



source: *ciemat.es*

- Developers:

ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

IPP Max-Planck-Institut für Plasmaphysik

- Collaborators:

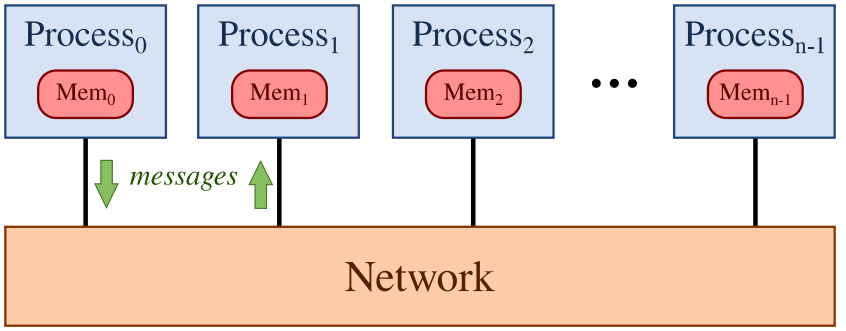
GOBIERNO DE ESPAÑA MINISTERIO DE CIENCIA E INNOVACIÓN

Ciemat Centro de Investigaciones Energéticas, Medioambientales y Tecnológicas

BSC Barcelona Supercomputing Center Centro Nacional de Supercomputación

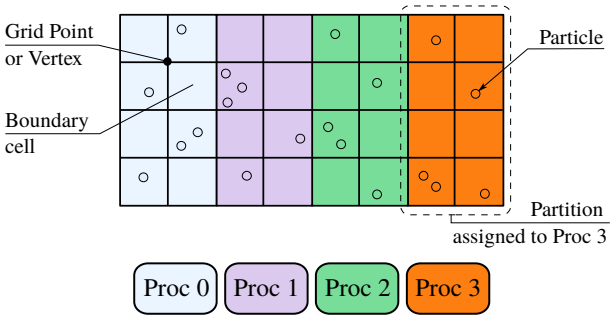
Particle in Cell

- Why to study them?
  - PIC is one of the most used methods in plasma physic simulations
  - PIC codes suffer from the defects of dealing with idealized configurations due to lack of computing power → process level parallelism
  - for new fusion reactors (ITER) → more physics and detail → need to adapt to new platforms
- Process Level Parallelism
  - Message Passing Model
  - Data decomposition can worsen the code performance



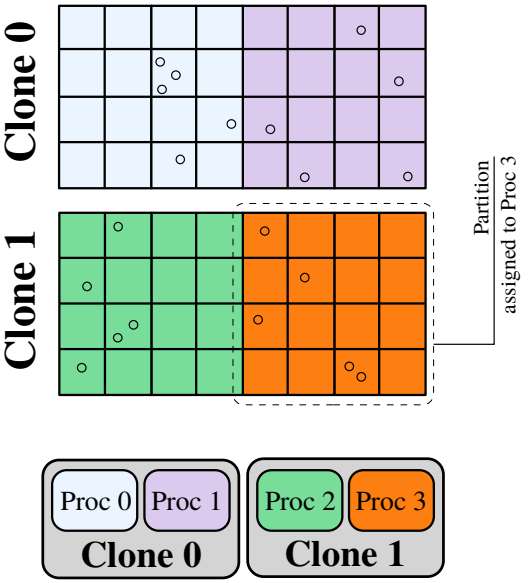
Process Level Parallelism

Domain Decomposition



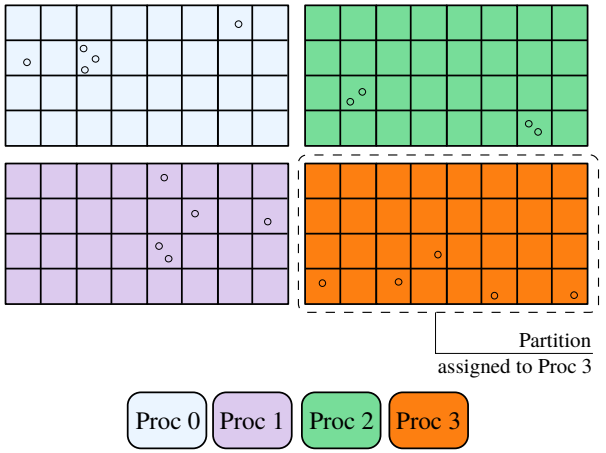
- ✔ Data locality
- ✖ Parallelization limited
- ✖ Unbalanced

Domain Cloning



- ✔ Scalability decoupled grid resolution
- ✔ Partial grid copy
- ✖ All-reduce communications

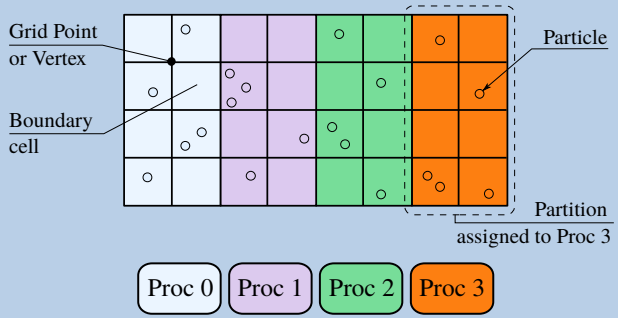
Particle Decomposition



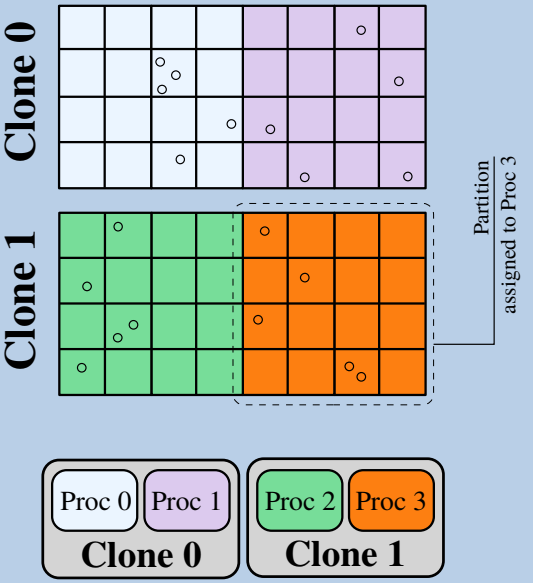
- ✔ Load balanced
- ✖ Expensive all-reduce
- ✖ Whole grid copy
- ✖ Not suitable for HPC

Process Level Parallelism

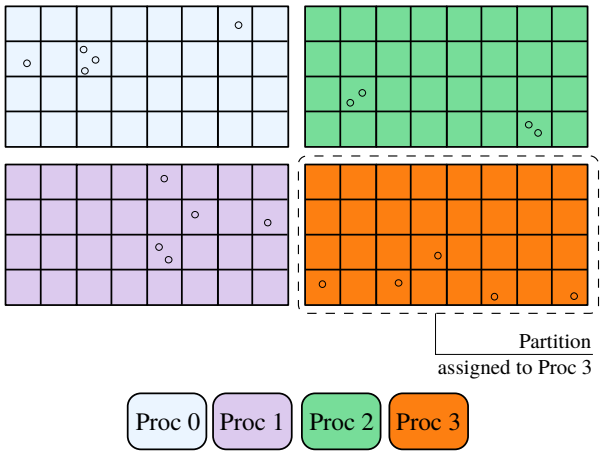
- Domain Decomposition



- Domain Cloning



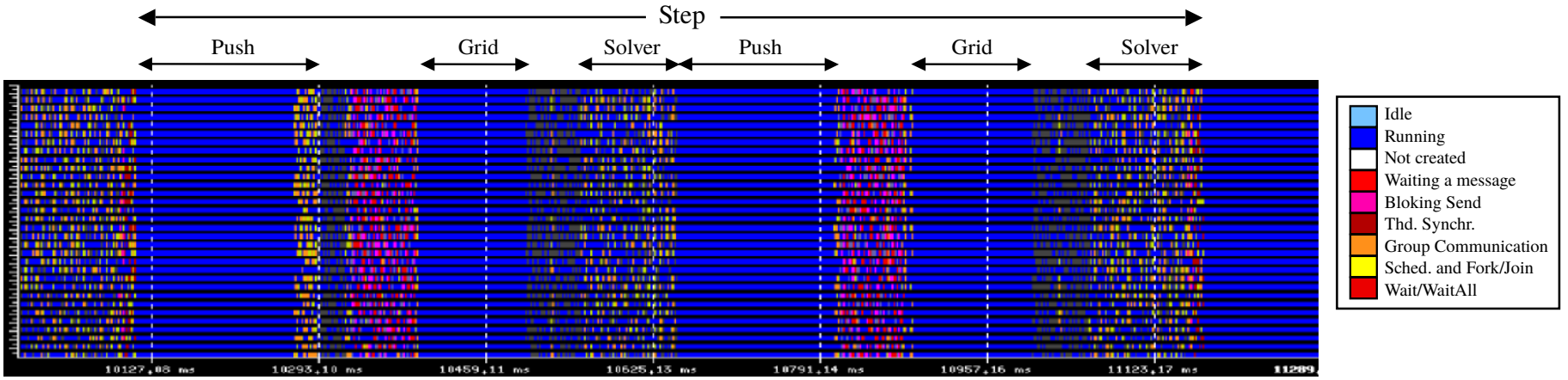
- Particle Decomposition



EUTERPE

Process Level Parallelism

- Load Balancing

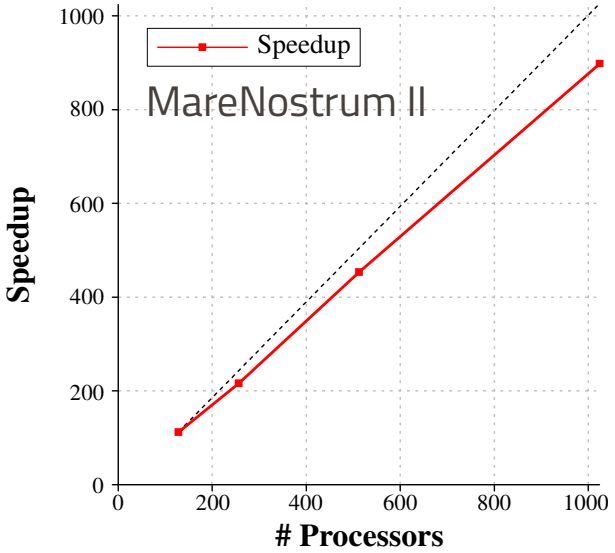


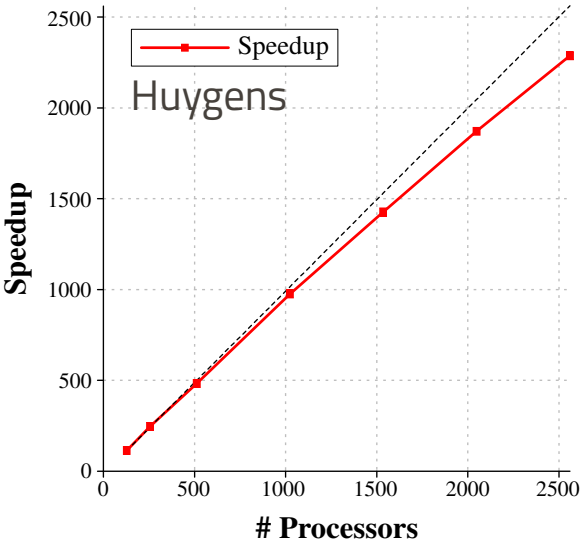
Domain Decomposition

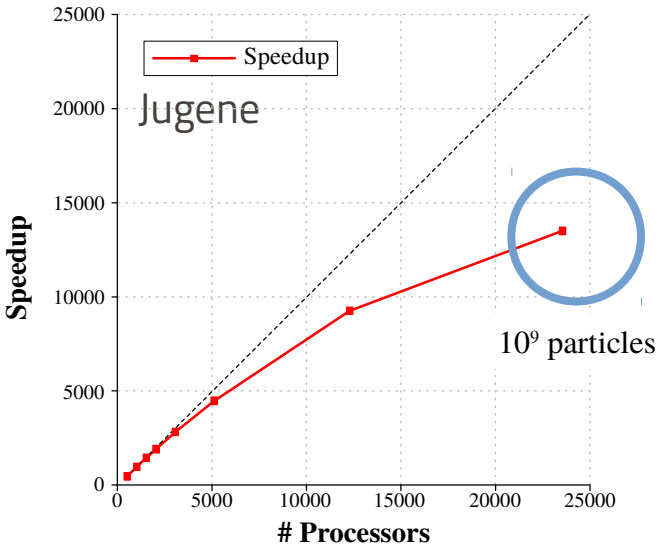
MareNostrum II

Process Level Parallelism

- Strong Scaling

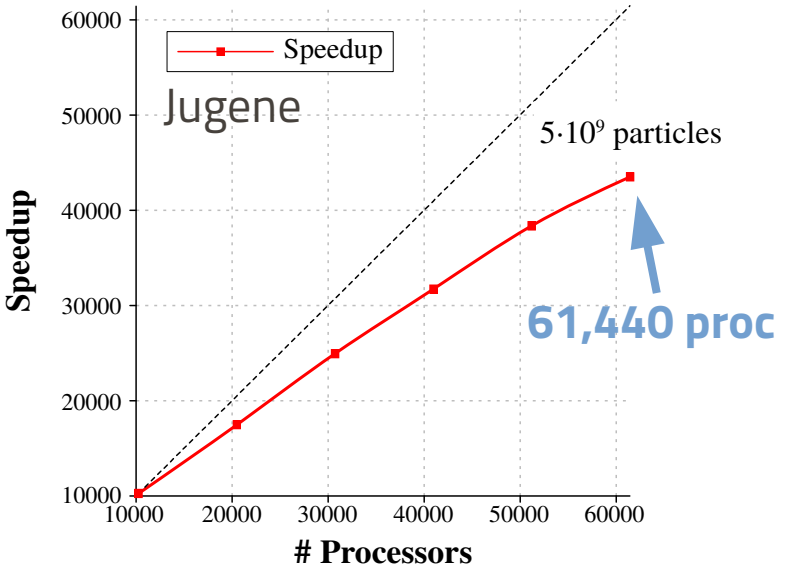






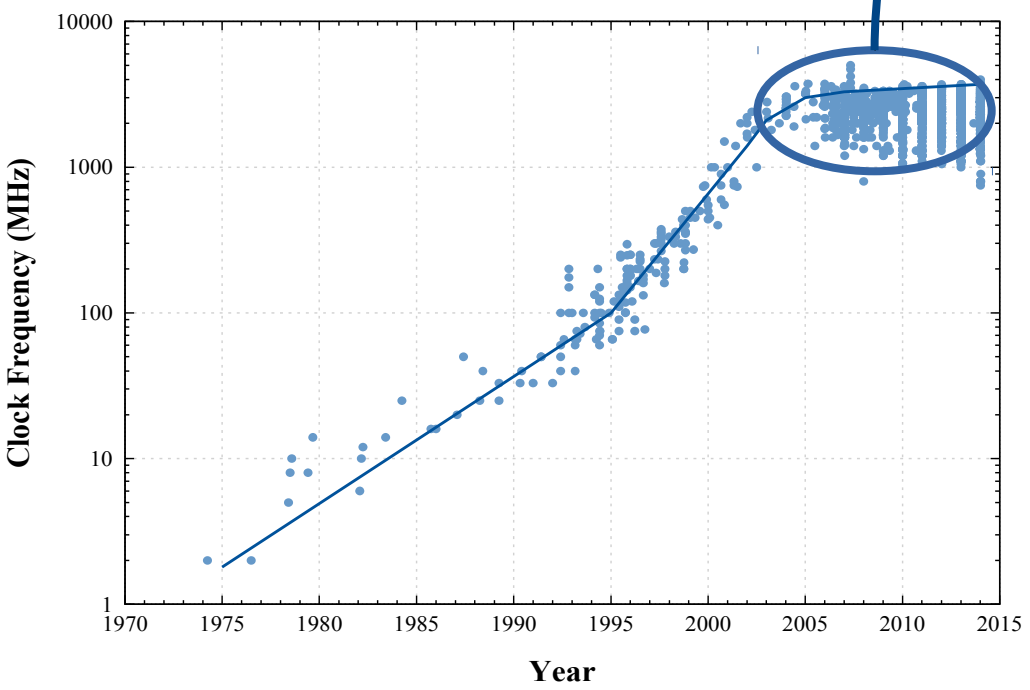
$$S(n) = \frac{T(1)}{T(n)}$$

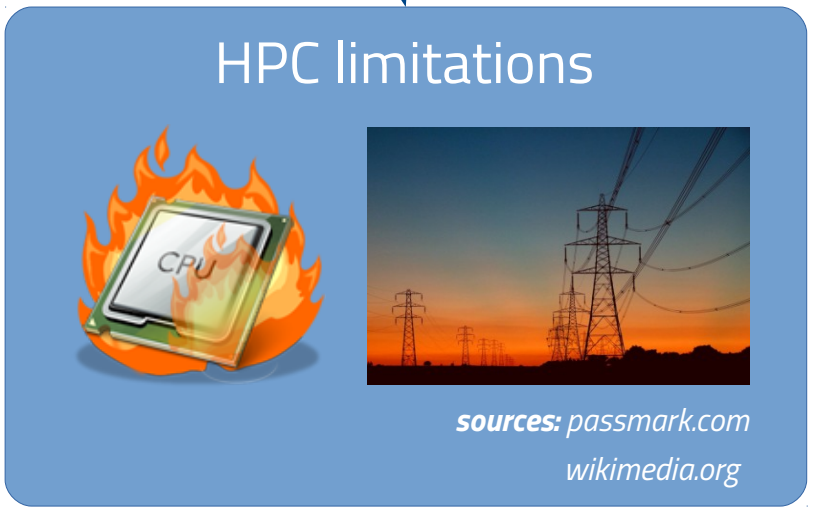
Domain Cloning

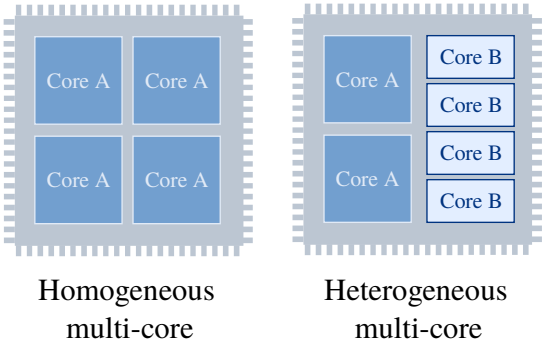


Multi-core Parallelism

- Evolution







Homogeneous multi-core

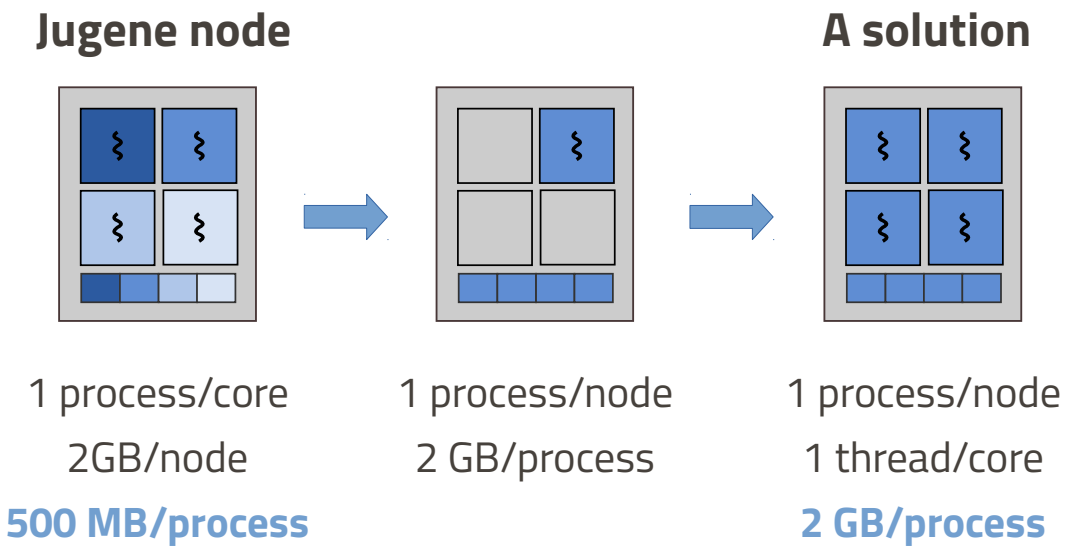
Heterogeneous multi-core

Multi-core Parallelism

- Influence on the codes
  - ✖ Limited space!

    ↓

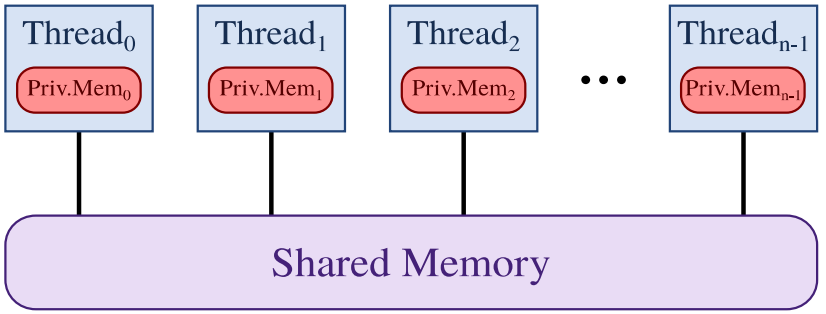
    Limited problems to solve or waste resources



- It is a need to adapt codes to new architectures to exploit all level of parallelism. Otherwise, we are paying for resources not used in HPC centers.

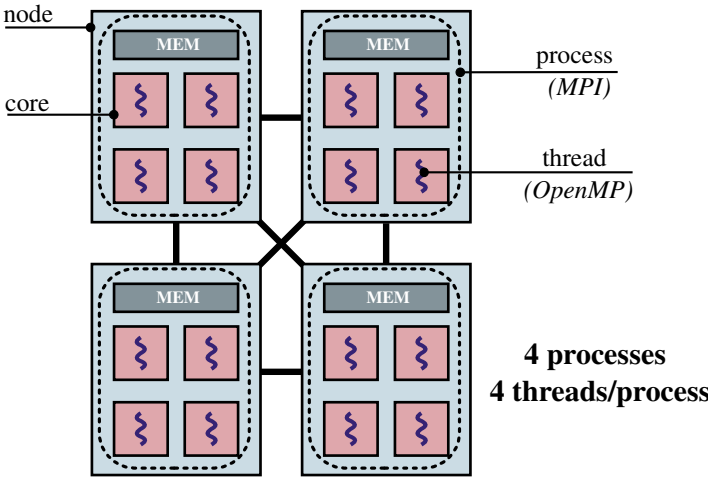
Thread Level Parallelism

- Shared Memory Model



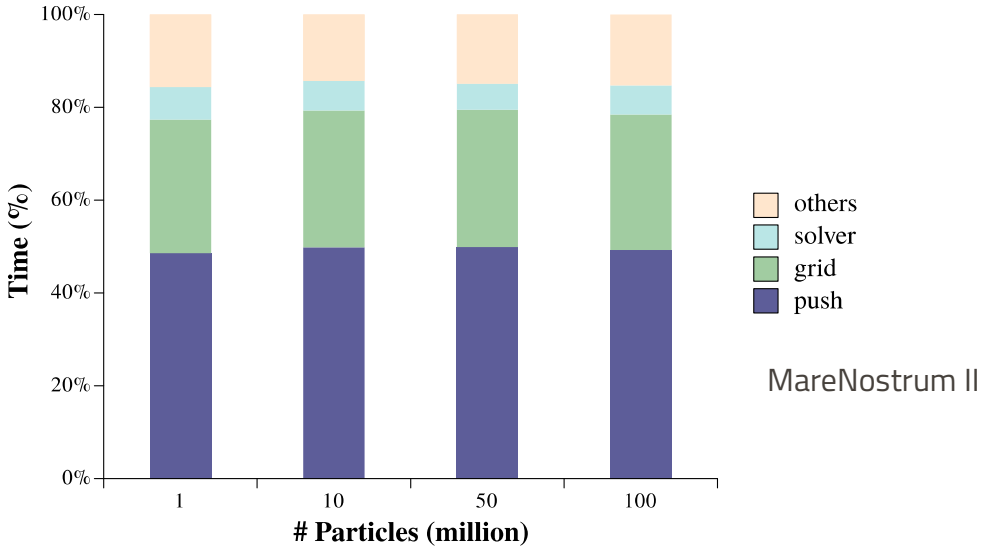
OpenMP

- Hybrid Version

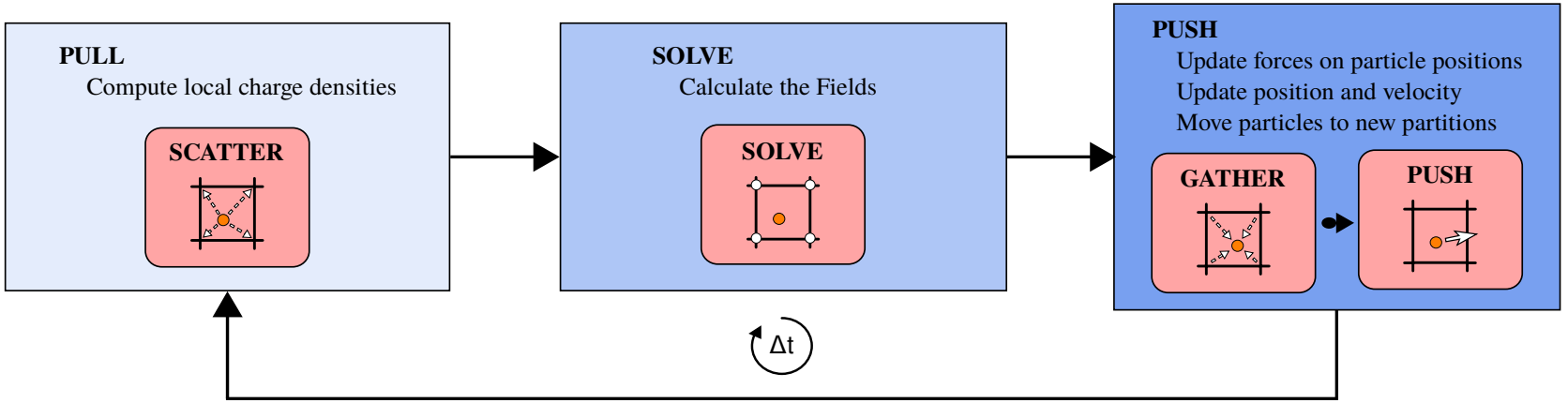


Thread Level Parallelism

- Time Distribution (EUTERPE)



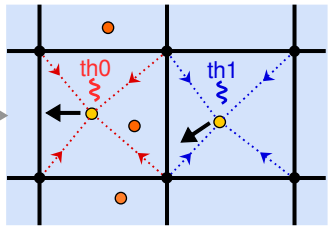
MareNostrum II

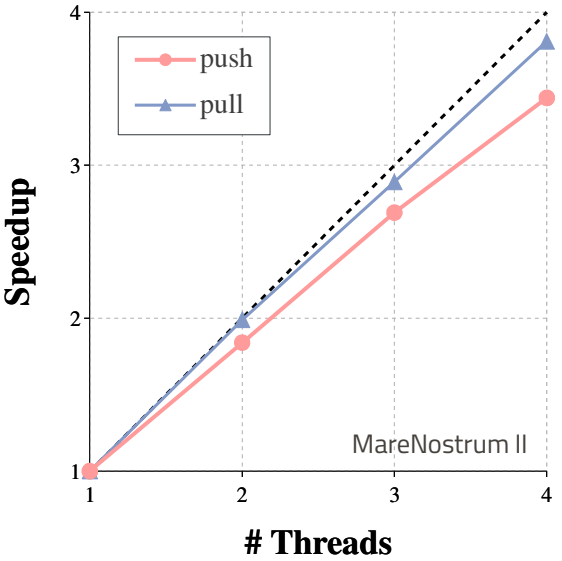


Thread Level Parallelism

- Push

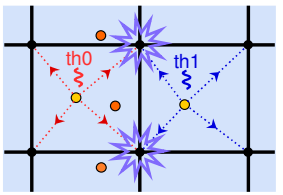
```
...
do ip = 1, npart_loc
   (rt,zt,phit) = pic_loc(ip)
   ...
   call getfield (rt,zt,phit...)
   ...
   if (diag_flag) then
      ...
   endif
enddo                          PARALLEL DO
...
```

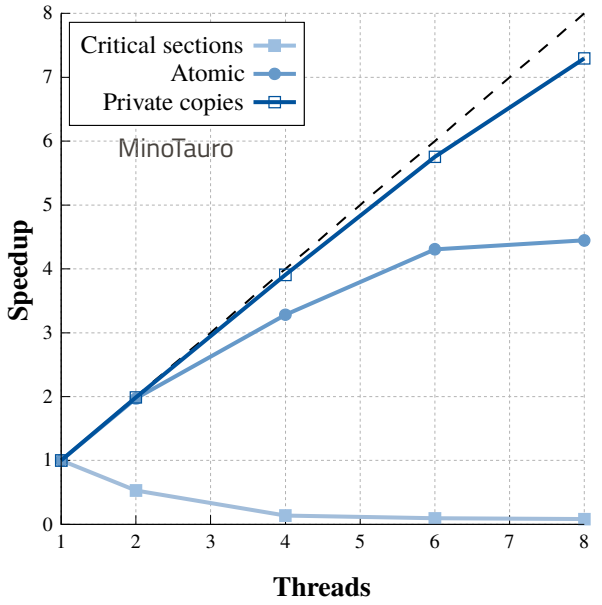




- Pull

```
rho = 0.0
...
do  ip = 1, npart_loc
   navg = gyroring_points(ip)
   ...
   do  l = 1, navg
      (s,chi,phi) = gridcell(ip,l)
      ...
      do  p = 1, 4
         (i,j,k) = gridvertex(s,chi,phi)
         rho(i,j,k) = rho(i,j,k) + ...
      enddo
   enddo
   ...
enddo                          PARALLEL DO
...
```





Thread Level Parallelism

- Pull Phase Code

```
numthrds = OMP_GET_MAX_THREADS()
ALLOCATE (rho_OMP(islw:isup, jclw:jcup, kplw:kpup, numthrds))
rho_OMP(:,:,:,:) = 0.0
rho(:,:,:) = 0.0
...
!$OMP PARALLEL DO                                                    &
!$OMP SCHEDULE (STATIC)                                              &
!$OMP DEFAULT  (NONE)                                                &
!$OMP PRIVATE  (ip, rt, zt, phit, vpart, wt, mut0, cosalp, pvol, s_gc,    &
!$OMP           chi_gc, b_abs, vperp2, u_bulk, ze_maxw, ztval_inv,        &
!$OMP           zvth2_inv, zf0, zphi_adi_val, za_par_adi_val, rhog, navg, &
!$OMP           wght, wght_cur, zcos1, zsin1, l, rt_x, zt_x, st_x,        &
!$OMP           chit_x, k, k_loc, wphi, phase, i, ws, j, wchi, kc, kkc,   &
!$OMP           temp1, jc, jjc, temp2, zksi, xksi, yksi, thrid )          &
!$OMP SHARED   (npart_loc, pic1_loc, pic2_loc, twopinfp, zreduce_flag,    &
!$OMP           sgrid, ns, nsel_ubulk, msdmp, turbo_adi, Phi_bspl, qsde,  &
!$OMP           Apar_bspl, zcurr_flag, species, msdqs, zcos2, zsin2,      &
!$OMP           dphigrid_inv, phigrid, k_offset, dsgrid_inv, chigrid,     &
!$OMP           dchigrid_inv, rho_OMP, b_equ, nequ_zm1,                   &
!$OMP           nequ_rm1, nequ_z, nequ_r, iphipr, iphipl, nphip_min,      &
!$OMP           rgedz, rgedr, rphi_map, phi_edge, gez_max, gez_min,       &
!$OMP           ger_max, ger_min, nphip, kplw, jclw, islw, smax0,         &
!$OMP           me_cart, nidbas, ds_profiles_inv, t_s, ubulk_s, n_s, elmag)

   DO ip = 1, npart_loc
      thrid = OMP_GET_THREAD_NUM() + 1
      rt     = pic1_loc(R_PIC,ip)
      zt     = pic1_loc(Z_PIC,ip)
      phit   = pic1_loc(PHI_PIC,ip)
      ...
      rho_OMP(i,jjc,kkc,thrid) = rho_OMP(i,jjc,kkc,thrid) + temp2*ws(j)
      ...
   END DO
!$OMP END PARALLEL DO
   ...
   DO ii = 1, numthrds
      rho(:,:,:) = rho(:,:,:) + rho_OMP(:,:,:,ii)
   ENDDO

   DEALLOCATE (rho_OMP)
```

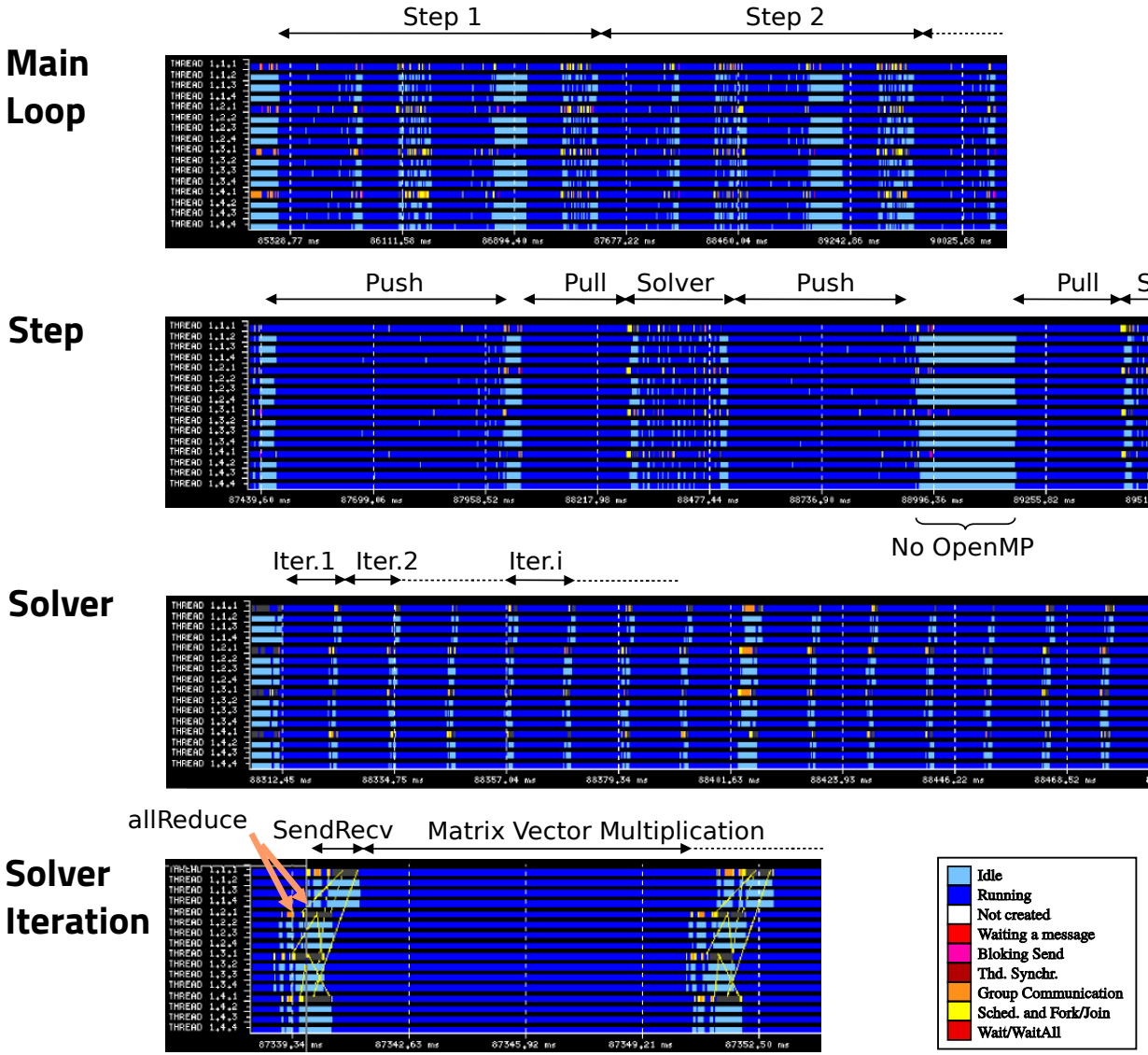
Thread Level Parallelism

- Solver Phase
  - Solve Linear Sparse Systems
  - PETSc issue → Not Thread-safe
  - Develop a thread-safe solver → JPCG and BJPCG

$$Ax = b$$

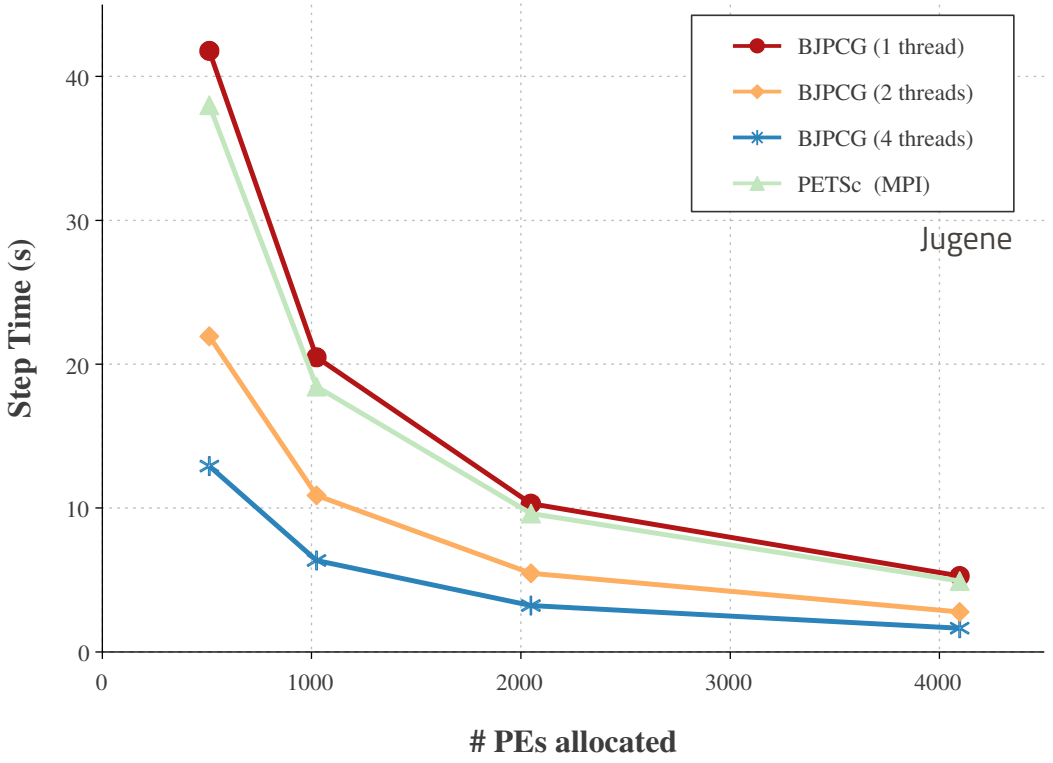
Thread Level Parallelism

- EUTERPE Hybrid version



Thread Level Parallelism

- EUTERPE Hybrid version

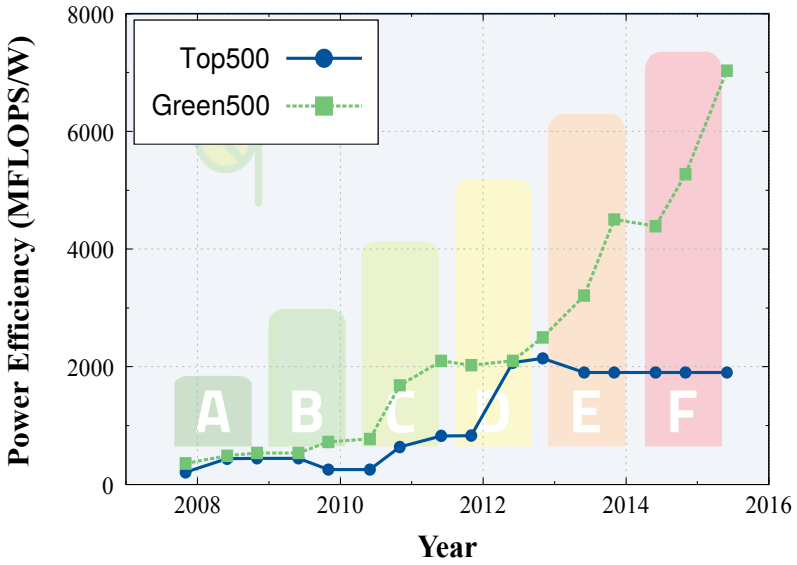


- 1,024 nodes
  - Original version can achieve 1,024 processors
  - Hybrid version can achieve 4,096 processors working with the same resources, so it improves the performance

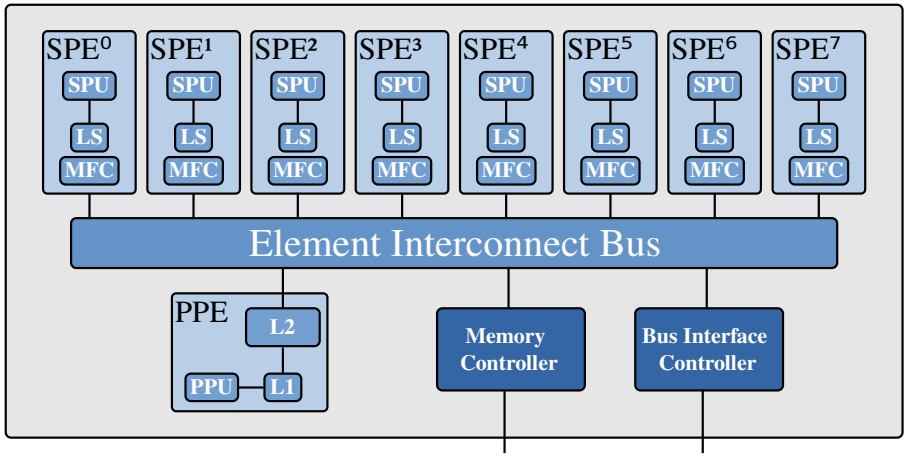
BJPCG:	128 blocks/process
Clone:	128 processes
Grid:	16x256x256
Particles:	10^9

Heterogeneous Parallelism

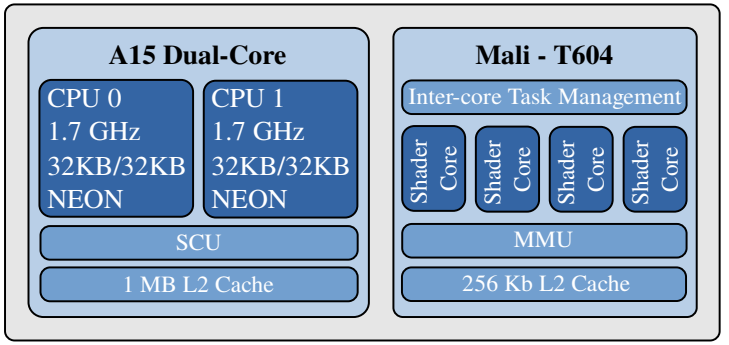
- Evolution



Cell (2009)



Samsung Exynos 5 Dual (2014)

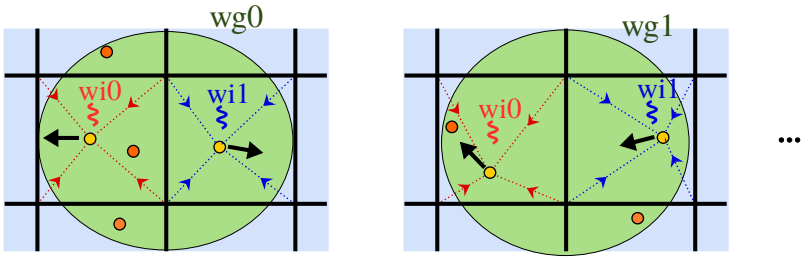


Heterogeneous Parallelism

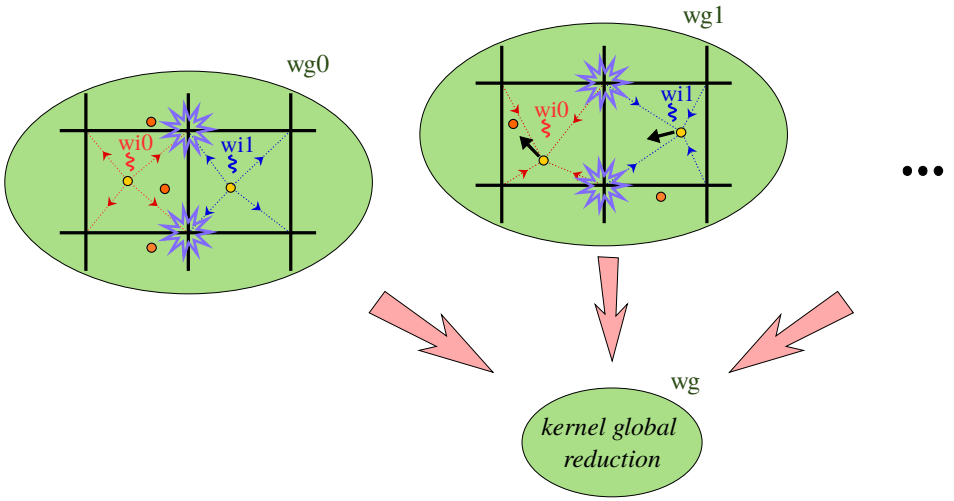
- Programming on Cell
  - Impressive peak performance → tedious programming process
    - Two different instruction sets (ISA)
    - DMA transfers → several alignment and size constraints
- ARM Mali GPU:
  - OpenCL (API) = improvement in the programming process
  - Unified memory system: local, private and global

Heterogeneous Parallelism

- Hybrid version (OpenMP + OpenCL)
  - OpenCL:
    - **Push**: similar OpenMP solution

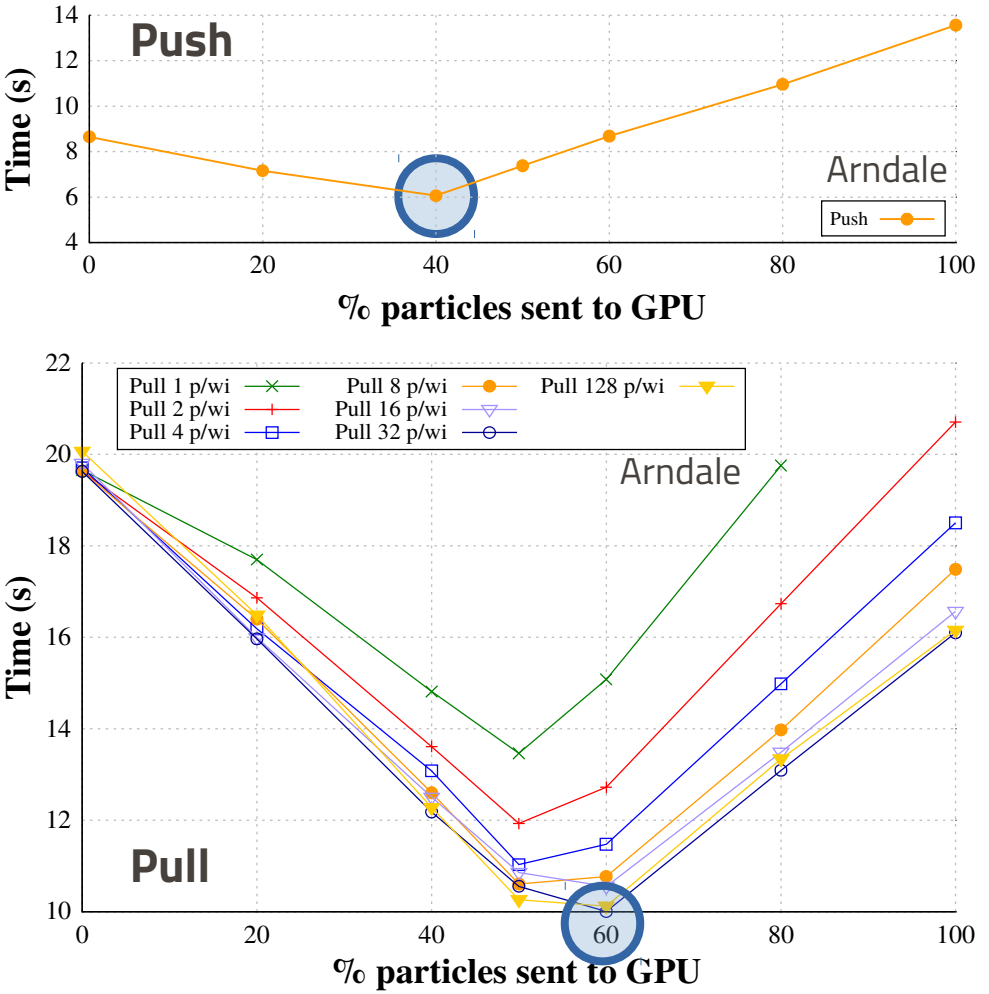


    - **Pull:** grid copy per work-group and atomics among work-items.



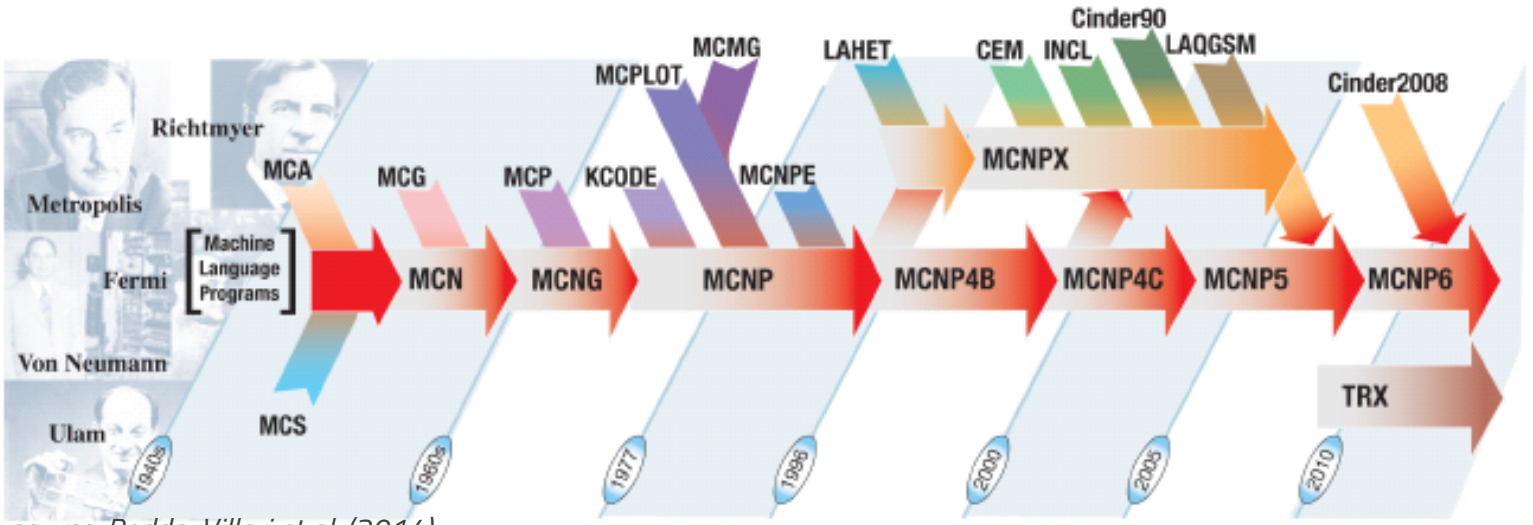
Heterogeneous Parallelism

- Hybrid version (OpenMP + OpenCL)
  - Hybrid version:
    - Static distribution of particles among devices (CPU and GPU)



Heterogeneous Parallelism

- But ...
  - Scientists are not in favor of porting production codes to new architectures:
    - It implies to re-validate the results
    - Codes are fruit of many collaborations



source: *Podda, Villari et al. (2014)*

MCNP6, fusion neutronics code developed from 1940s)

- A good solution could be a programming model that provides a certain level of abstraction over the hardware.

Heterogeneous Parallelism

- OmpSs (BSC)
  - Provides an abstraction that unifies SMP and heterogeneous programming in one model to reduce programmer effort
  - Taskification consists to reorganize some code to create taks (*groups of particles*).
- Push Phase

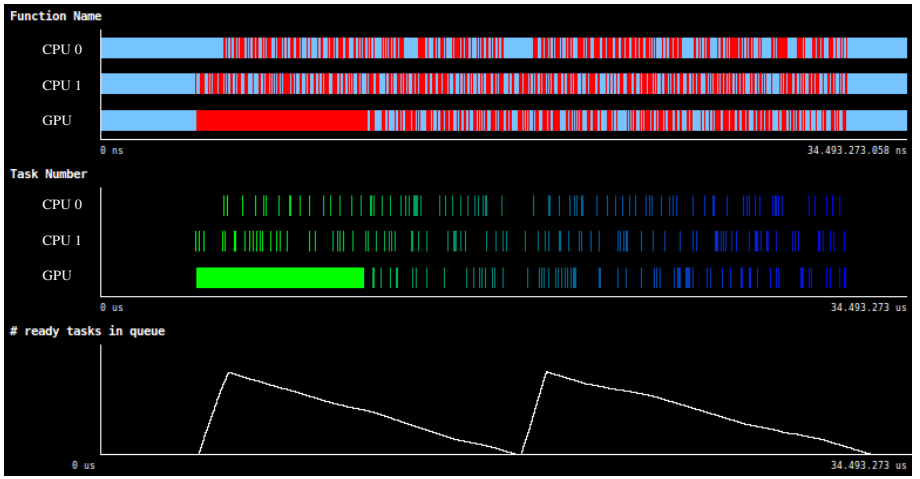
```
INTERFACE
   !$OMP copy_inout  (work1_loc(1:MAX_WORK, 1:(sup-inf+1)))    &
   !$OMP copy_in (pic1_loc, pic2_loc, msdqs, Phi_bspl,         &
   !$OMP          E_s_bspl, E_chi_bspl, E_phi_bspl, qsdms,     &
   !$OMP          msdmp, b_equ,sgrid, chigrid, phigrid,        &
   !$OMP          zcos2, zsin2, phieq, t_s, nsn_s, tst_s, n_s)

   !$OMP task  label(push_opencl)
   SUBROUTINE kernel_push (pic1_loc, pic2_loc, work1_loc,      &
   ...

   !$OMP target device(smp) implements (kernel_push)           &
   !$OMP copy_inout  (work1_loc(1:MAX_WORK, 1:(sup-inf+1)))    &
   !$OMP copy_in (pic1_loc, pic2_loc, msdqs, Phi_bspl,         &
   !$OMP          E_s_bspl, E_chi_bspl, E_phi_bspl, qsdms,     &
   !$OMP          msdmp, b_equ, sgrid, chigrid, phigrid,       &
   !$OMP          zcos2, zsin2, phieq, t_s, nsn_s, tst_s, n_s)

   !$OMP task  label(push_smp)
   SUBROUTINE kernel_push_smp (pic1_loc, pic2_loc, work1_loc,  &
   ...
END INTERFACE
```

```
CALL kernel_push (pic1_loc, pic2_loc, work1_loc(1,ipe),        &
                  species, dtloc, msdqs, qsdms, msdmp, ...
```



- Conclusion

Kernel	Performance - Best time (s)		Programmability - Productivity	
	OpenMP+OpenCL	OmpSs	OpenCL calls	OmpSs Directives
Push	6.02	6.92	161	12
Pull	10.31	10.83	167	18

- It is feasible to port a production PIC code to a heterogeneous platform
- OmpSs version is a bit slower but simpler → productivity ↑

Outline

Introduction to Plasma Physics

Supercomputing for Fusion Energy Applications

Particle-in-Cell Codes

- Fusion Group

Fusion Group

- Members
  - Mervi Mantsinen
  - Felipe Nathan de Oliveira
  - Shimpei Futatani
  - Dani Gallart
  - Albert Gutiérrez
  - Allah Rakah
  - Carles Riera
  - Xavier Sáez
- Research
  - Optimization of fusion applications
  - Understanding and control of ELM physics
  - Modelling of Ion Cyclotron Resonance Frequency Heating (ICRF)
  - Modelling DEMO reactor
  - Magnetohidrodynamics (MHD) and instabilities
  - Fusion neutronics

Fusion Group

- Collaborations

- More information
  - Blog: fusion.bsc.es (under construction)
  - Web: www.bsc.es
  - E-mail: {mervi.matsinen, shimpei.futatani} @ bsc.es

Thanks for your attention!